THESIS

SUPPLEMENTARY
MATERIAL

This is to certify that the

thesis entitled

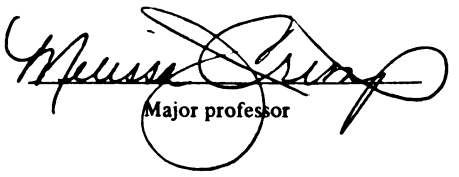
THE DYNAMICS AND SCIENTIFIC VISUALIZATION
FOR THE ELECTROPHORETIC DEPOSITION PROCESSING
OF SUSPENDED COLLOIDAL PARTICLES
ONTO A REINFORCEMENT FIBER

presented by

Peter Timothy Robinson

has been accepted towards fulfillment
of the requirements for

Masters degree in Engineering



Major professor

Date 7-16-93

O-7639

MSU is an Affirmative Action/Equal Opportunity Institution

LIBRARY
Michigan State
University

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
MAY 11 2003		

MSU Is An Affirmative Action/Equal Opportunity Institution
c:\circ\datedue.pm3-p.1

THE DYNAMICS AND SCIENTIFIC VISUALIZATION FOR THE ELECTROPHORETIC DEPOSITION PROCESSING OF SUSPENDED COLLOIDAL PARTICLES ONTO A REINFORCEMENT FIBER

By

Peter Timothy Robinson

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Material Science and Mechanics

1993

T
FOR

To meet

niques of matri

been introduced

pended colloida

been developed

ises numerical

icles are due to

Stoke's drag.

Visualiz

computer. A me

condition param

matrix - collecto

The coll

sters for Fe-40

the foundation

simulator.

ABSTRACT

THE DYNAMICS AND SCIENTIFIC VISUALIZATION FOR THE ELECTROPHORETIC DEPOSITION PROCESSING OF SUSPENDED COLLOIDAL PARTICLES ONTO A REINFORCEMENT FIBER

By

Peter Timothy Robinson

To meet the demands for new, innovative and more efficient manufacturing techniques of matrix composite materials, a method based on the ideas of colloid science has been introduced. The method relys on maximizing the electrophoretic deposition of suspended colloidal matrix particles onto a reinforcement fiber. A numerical algorithm has been developed to simulate the many body problem for the colloidal system. The algorithm uses numerical integration to solve the dynamical equations of motion. Motion of the particles are due to London - van der Waal forces, Coulombic forces, gravitational forces and Stoke's drag.

Visualization of the algorithm in two dimensions has been attempted on a personal computer. A menu user interface allows flexibility and efficiency for modifying the initial condition parameters such that the optimal initial condition parameters that maximize the matrix - collector deposition may be determined.

The colloidal suspension simulator algorithm was intended to be tested with parameters for Fe-40Al matrix particles using an Al_2O_3 reinforcement fiber. This thesis presents the foundation work necessary for the construction of a functioning colloidal suspension simulator.

TABLE OF CONTENTS

LIST OF TABLES

Figure

1.1 Fiber
The
throu
the fi
and F
40A.
the id

2.1 Free
This
partic
F_{Db}
F_{Ste}
F_{B-i}
Vel.
F_{Str}
F_{Lo}
F_{Gra}
F_{Hy}

2.2 Gou
This
dal p

2.3 Ratio
The
have
Mys
Krie

2.4 The
This
repre
as +
of th
parti
Mys
Krie

LIST OF FIGURES

2.5 Mole
This
spher
Ham
van d

3.1 Flow
This
throu
bodie

3.2 Grap
This
and y
partic
of pa

4.1 Comm
This f
RUN

5.1 Zeta p
The m
volum

5.2 Zeta p
The m
umeC

5.3 Zeta p
The m
umeC

5.4 Zeta po
The me
dispers
Wilson

5.5 Zeta po
The me
powder
by Bret

5.6 Data file
This figu
CSSRU

gram. Th
the radi
rest of th
lowed by

Depictio
This fig
using th

A	Hamaker
q	Electric p
e	Electroni
ε	Dielectric
F_{DEL}	Coulomb
F_g	Force due
F_{HYD}	Hydrody
F_{TOT}	Total forc
F_{VDW}	London -
V	Electric p
k	Boltzman
T	Temperat
V_R	Repulsive
z	Charge de
n_0	Bulk con
V_0	Electric p
V_s	Electric p
ζ	Electric p
κ	Debye-H
n_i	Concentra
n_0	Bulk conc
η	Viscosity
u	Electroph
A	London -
x_m	x position
y_m	y position
v_m	x compon
v_m	y compon
z	Valence n

LIST OF SYMBOLS

A	Hamaker constant
q	Electric point charge
e	Electronic charge = 1.60217733E-19 Coulomb
ε	Dielectric constant
F_{DBL}	Coulombic, double layer force
F_g	Force due to gravity
F_{HYD}	Hydrodynamic force
F_{TTL}	Total force acting on a colloid
F_{van}	London - van der Waals force
ψ	Electric potential
k	Boltzmann constant = 1.380658E-23 J/K
T	Temperature
V_R	Repulsive potential energy
ρ	Charge density
n_o	Bulk concentration of positive and negative ions
ψ_o	Electric potential at the surface of a colloid
ψ_s	Electric potential at the Stern layer
ζ	Electric potential at the surface of shear; the zeta potential
κ	Debye-Hückel reciprocal length parameter
n_i	Concentration of ions
n_o	Bulk concentration of an ionic species
η	Viscosity
μ	Electrophoretic mobility or mean
Λ	London - van der Waals constant
x_{xi}	x position
x_{yi}	y position
v_{xi}	x component of velocity
v_{yi}	y component of velocity
z	Valence number of an ion

The manuf

e ideas of colloi

oretic deposito

ber is pulled thro

les adhere to the

major advantage

matrix particles c

composite materi

resistance to fati

To unders

colloid chemistry

that lie between

between the mic

and the macrosc

A compl

process involves

are transported

with the collecto

close enough to

into the nature o

classical theore

INTRODUCTION

The manufacturing of an intermetallic matrix composite using a process based on the ideas of colloid science has been proposed [1]. The proposed process involves electrophoretic deposition of Fe-40Al matrix particles onto a bundled Al_2O_3 fiber. The bundled fiber is pulled through a concentrated suspension of the particles, Figure 1.1, and the particles adhere to the fiber due to adhesive, London - van der Waal and Coulombic forces. A major advantage of this technique over other production methods is the uniformity of matrix particles covering the fiber. The uniformity produced by this process results in a composite material that theoretically has improved mechanical properties and has greater resistance to fatigue.

To understand the electrophoretic deposition process, investigation into the field of colloid chemistry is required. Physically, the colloidal domain is the size range of particles that lie between one nanometer and one micron. The domain of colloid chemistry lies between the microscopic size range, in which the strong and weak nuclear forces dominate and the macroscopic size range, in which gravitational forces dominate.

A complete description of the dynamics of the colloidal electrophoretic deposition process involves two steps. The first step is the transport step in which the colloid particles are transported through the suspension medium and come into contact with each other or with the collector. The second step is the surface interaction step in which the particles are close enough to each other or to the collector such that surface interactions occur. Inquiry into the nature of the colloidal interactions between the surfaces of the colloids leads to the classical theoretical description developed independently by Derjaguin and Landau, and by


Al_2O_3
Fiber

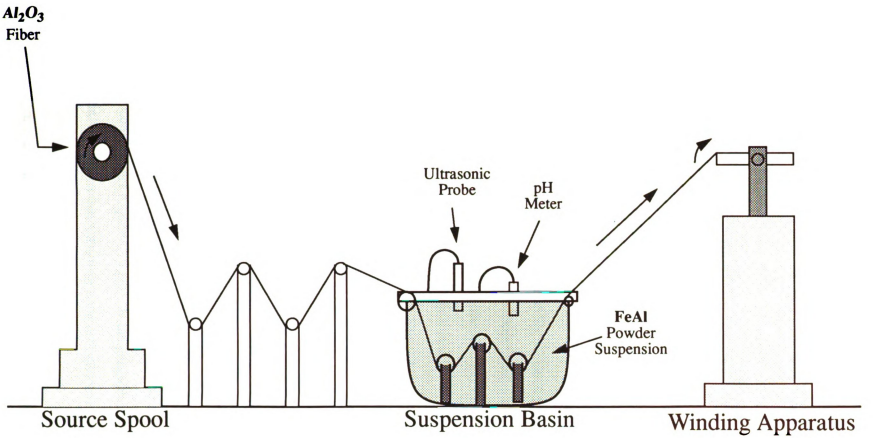


Figure 1.1 Fiber electrophoretic deposition Processing of Fe-40Al/Al_2O_3.

The schematic diagram shows the FeAl fiber being pulled through the suspension basin. The colloidal particles adhere to the fiber by maximizing the attractive forces between particle and fiber and by minimizing the homocoagulation between Fe-40Al - Fe-40Al and Al_2O_3 -Al_2O_3. This schematic shows only the idea of the production method.

Verwey and Overb

suggests that the st

action energy actu

by a colloid is the

repulsive energy.

London-van der W

Coulombic electr

system will be de

Several va

process. Changes

ticle size or the die

a composite that

posed with diffe

the DLVO theor

model.

The com

mented as a com

persion behavio

may be initializ

run. For the init

tor over a time

a specified time

process for a se

any initial cond

dict optimal ini

set of composit

The pro

Verwey and Overbeek [2,3] referred to in literature as the DLVO theory. The DLVO theory suggests that the stability of a colloidal suspension is determined by the total surface interaction energy acting between the colloids. The total surface interaction energy possessed by a colloid is the sum of the electrodynamic attractive energy plus the electrodynamic repulsive energy. The electrodynamic attractive energy is the direct consequence of the London-van der Waal attractive force. The electrodynamic repulsive energy arises from the Coulombic electrostatic force. Other possible forces that may also be present in a colloidal system will be described in greater detail later in this thesis.

Several variables dictate the quality and efficiency of the electrophoretic deposition process. Changes in the processing pH level, the initial electrolyte concentration, the particle size or the dielectric and Hamaker constants for different types of materials can lead to a composite that is more or less uniformly distributed than a composite that has been processed with different starting conditions. This situation and the question of how accurately the DLVO theory models the physical world has lead to the construction of a computer model.

The computer model is the focus of this thesis. The model is an algorithm implemented as a computer program and designed to provide scientific visualization of the dispersion behavior of the composite components while they are in suspension. The program may be initialized with information for both matrix particles and collector and allowed to run. For the initial conditions provided, a measurement of developed mass onto the collector over a time period may be recorded. The recorded developed mass on the collector for a specified time period indicates a measure of the success of the electrophoretic deposition process for a set of initial conditions. The program is flexible to allow the user to change any initial conditions that are required. A goal for the program is that it will be able to predict optimal initial conditions that maximize the electrophoretic deposition for an arbitrary set of composite components.

The program uses a numerical algorithm to simulate the many body problem for

suspended colloi

the suspended pa

-corrector" meth

ger of the algori

algorithm, the fo

motion are derive

In order t

ex surface poten

must be characte

ron aluminide p

mental approxim

potential. Using

dynamics of the e

well the algorith

insight into the w

This thesi

ing a colloidal sus

form on a 3 1/2 i

suspended colloidal particles. The dynamical equations of motion describe the motion of the suspended particles as time evolves. The numerical algorithm is based on a "predictor - corrector" method for numerically integrating the equations of motion. For the "predictor" part of the algorithm, the Euler modified method is used. For the "corrector" part of the algorithm, the fourth order Adams - Bashforth method is used. The dynamical equations of motion are derived by incorporating the total forces that act on the system of particles.

In order to obtain the total surface energy described by the DLVO theory, the electric surface potential of the matrix particles and the electric surface potential of the collector must be characterized. Electrokinetic sonic amplitude (ESA) measurements were made for iron aluminide powder and for alumina fiber. The ESA measurements provide the experimental approximation to the electric surface potential in the form of the measured zeta potential. Using the zeta potential data as an input into the computer model, analysis of the dynamics of the electrophoretic deposition process may be conducted. Questions as to how well the algorithm describes the physical world will be addressed in order to gain better insight into the workings of the electrophoretic deposition phenomenon.

This thesis provides the algorithm necessary to generate a computer tool for analyzing a colloidal suspension. A semi functioning computer program is included both in binary form on a 3 1/2 inch floppy disk and in printed form in the appendices.

A colloid

from 1.0 x 10^{-6} m

roscopic objects

examples of colloi

butter, jellies, star

general, a colloid

dispersion media

phase in a liquid

tem that has a liq

Table 2.1 outline

The class

as a colloidal sus

gation, the disper

be deionized, dis

suspension is ess

tation process.

To attemp

many contributin

ticle size is distri

dence correlating

important factor.

other geometrical

tors to be conside

and the amount of

LITERATURE REVIEW

A colloid is a particle that has at least one of its three dimensions in the size range from 1.0×10^{-9} meters to 1.0×10^{-6} meters. The branch of science that studies these macroscopic objects is called colloid chemistry. D.H. Everett [4] discusses several familiar examples of colloidal systems including the following: fogs, mists, tobacco smoke, milk, butter, jellies, stained glass, photographic "emulsions", blood, paints, muds and slurries. In general, a colloidal system is composed of a disperse phase; a gas, solid or liquid, and a dispersion medium; a gas, solid or liquid. A colloidal system that has a liquid disperse phase in a liquid dispersion medium, for example, is termed an emulsion. A colloidal system that has a liquid disperse phase in a gas dispersion medium is called a liquid aerosol. Table 2.1 outlines several types of colloidal systems.

The class of colloidal system in which a solid is dispersed in a liquid is referred to as a colloidal suspension or a sol. For the electrophoretic deposition process under investigation, the disperse phase will be iron aluminide particles and the dispersion medium will be deionized, distilled water. Therefore, an investigation into the dynamics of a colloidal suspension is essential in order to formulate a physical model of the electrophoretic deposition process.

To attempt a complete description of the overall dynamics of a colloidal suspension, many contributing factors need to be considered. The particle size as well as how the particle size is distributed are two such factors. Wiese and Healy [5] found experimental evidence correlating the particle size with colloid stability. The particle shape is another important factor. Intuitively, the motion of a spherical particle will behave unlike some other geometrically shaped object, such as a cube or needle, while in suspension. Other factors to be considered are the particles surface properties, such as the surface composition and the amount of electrical charge on the particles surface Finally, the primary factor con-

Table 2.1 C

Everett, D. H., B
1988

Disper Phase
Liquid
Solid
Liquid
Solid
Solid
Solid
Gas

Table 2.1 Colloidal Systems.

Everett, D. H., Basic Principles of Colloid Science, Royal Society of Chemistry, London, 1988.

Disperse Phase	Dispersion Medium	Class	Examples
Liquid	Gas	Liquid aerosol	Fog, mist, tobacco smoke, "aerosol" sprays
Solid	Gas	Solid aerosol	Industrial smokes
Liquid	Liquid	Emulsions	Milk, butter, may-onnaise, asphalt
Solid	Liquid	Colloidal Suspension	Silver iodide, paints
Solid	Liquid	Paste	Clay slurries, tooth-paste, muds
Solid	Solid	Solid Suspension	Opal, pearl, stained glass
Gas	Liquid	Foam	Froths, foams

tributing to the dynamics of a colloidal suspension is the total force acting on an individual colloid particle. The total force can be derived from knowledge of the particles size, shape and surface properties as well as from the dispersion medium properties.

Several forces collectively define the total force acting on an individual colloid particle. Electrodynamic forces include the attractive London-van der Waals force and the repulsive electrostatic double layer force. Hydrodynamic forces, that obey Stoke's law, arise from viscosity and are proportional to the velocity of the particle moving in the dispersion medium. Steric forces are repulsive forces that may be present from the overlap of adsorbed polymer layers. Brownian motion is erratic particle motion driven from fluctuations in the density of the liquid. Structural forces are strong repulsive forces that act over a very short range and result from changes in the dispersion medium structure in the vicinity of the surface or interface [6,7]. The gravitational force is an attractive force obeying Sir Isaac Newton's law of gravitation. Lastly, the presence of an external electric or magnetic field can affect the motion of a charged colloid particle. Table 2.2 outlines the possible forces that may be present in a colloidal system. Figure 2.1 shows a free body diagram for a suspended colloidal particle.

Electrodynamic forces, comprised of the repulsive double layer overlap force and the attractive London-van der Waal force, are present only if the colloid particles have electrically charged surfaces. Although some special types of colloidal suspensions exist in which the colloid particles possess no surface charge, the majority of colloidal suspensions do, in fact, contain electrically charged particles.

A colloid in suspension can obtain a surface charge through several mechanisms. Many of these mechanisms are described by Ross and Morrison and by Hirtzel and Rajagopalan [2,6] and include preferential adsorption of ions, accumulation of electrons at the interface and adsorption of polyelectrolytes.

Realizing that suspended colloids possess a surface charge led to the idea of the electrostatic double layer. The double layer concept, originated by Helmholtz (1879) and

Table 2.2 Forc

The table shows th
and lists the variab
Russel, William B
Press, 1987.

F
Electrodynamic - London - va - Electrostati
Hydrodynamic - Stokes Dra
Steric
Brownian
Structural
Gravitation
External Field - Magnetic - Electric

Table 2.2 Forces in a colloidal system.

The table shows the possible forces that may be present in a colloidal suspension system and lists the variables on which the forces depend.
Russel, William B., The Dynamics of Colloidal Systems, The University of Wisconsin Press, 1987.

FORCE	FUNCTION OF
Electrodynamic	
- London - van der Waals	Displacement, Material, Surface Charge
- Electrostatic (Double Layer)	Displacement, Material, Surface Charge
Hydrodynamic	
- Stokes Drag	Viscosity, Velocity
Steric	Displacement (Position of adsorbed polymer layers)
Brownian	Thermal Motion (Density Fluctuations)
Structural	Displacement
Gravitation	Mass, Displacement
External Field	
- Magnetic	Surface Charge, Displacement, Field Strength
- Electric	Surface Charge, Displacement, Field Strength

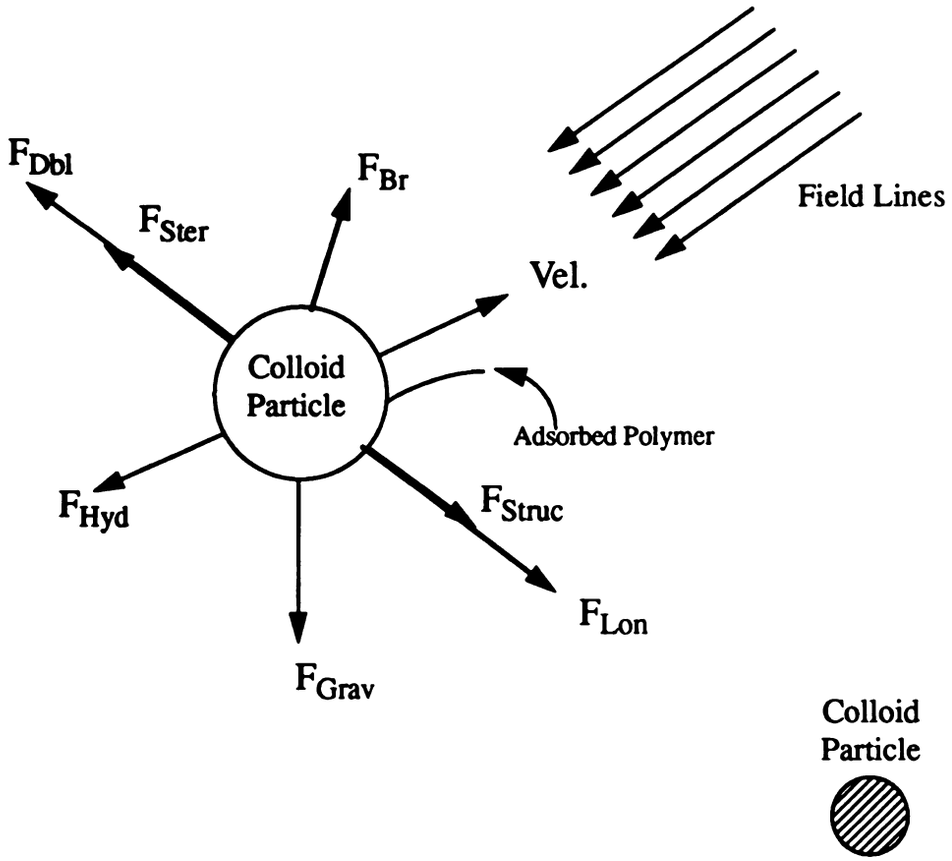


Figure 2.1 Free body diagram for a colloidal particle in suspension.

This figure shows possible force vectors acting on the colloid particle, where,

F_{Dbl}	is the double layer force,
F_{Ster}	is the steric repulsive force,
F_{Br}	is the force due to Brownian motion,
Vel.	is the velocity vector of the particle,
F_{Struc}	is the structural force,
F_{Lon}	is the attractive London - van der Waals force,
F_{Grav}	is the force due to gravity,
F_{Hyd}	is the hydrodynamic force.

vestigated in gr

;93b, suggests

fashion. The fixe

as counterions

or the surface of

the particle creat

2.2 shows an exp

The cloud

charge on the su

fixed surface ch

absence of therm

and would comp

situation would

formly distribute

colloids surface.

other counterion

the charged ions

Before a

other important

the electrical po

defined as the en

the electrical dou

to be electric poi

ctent Coulombic

investigated in greater detail by Louis George Gouy (1910) and David Leonard Chapmann (1913), suggests that a cloud of ions gather around a suspended colloid in an organized fashion. The fixed charge on the colloids surface attracts free ions of opposite sign, referred to as counterions. The attracted counterions, in turn, attract ions of the same sign as the ions on the surface of the colloid, called coions. This cycle repeats outward from the surface of the particle creating integrated layer upon layer of attracted coions and counterions. Figure 2.2 shows an exploded view of the Gouy-Chapman layer surrounding a particle.

The cloud of diffuse ions that surround the colloid exactly neutralizes the fixed charge on the surface of the colloid. Collectively, the diffuse ion region and the colloids fixed surface charge region make up what is called the electrical double layer. In the absence of thermal agitation, counterions would migrate to the surface of a charged particle and would completely cover it, exactly neutralizing its charge [8]. The double layer in this situation would be extremely compact. In reality, thermal motion has a tendency to uniformly distribute the free ions in the dispersion medium. As counterions are attracted to the colloids surface, the counterions produce a screening effect that blocks further attraction of other counterions. The combined effects of thermal agitation and screening of ions causes the charged ions in the diffuse region of the double layer to have a distinct distribution.

Before a description of how the charged ions in the diffuse region are distributed, other important characteristic parameters of the double layer need to be defined, namely, the electrical potential energy and the electric potential. Potential energy, in general, is defined as the energy that a system possesses as a result of its configuration. In the case of the electrical double layer, ions and counterions that surround the colloid may be assumed to be electric point charges. These electric charges are separated by varying distances and exert Coulombic forces upon each other of the form:

$$F = \frac{q_i q_j}{\varepsilon r^2}, \tag{1}$$

Figure 2

This fig
ions and

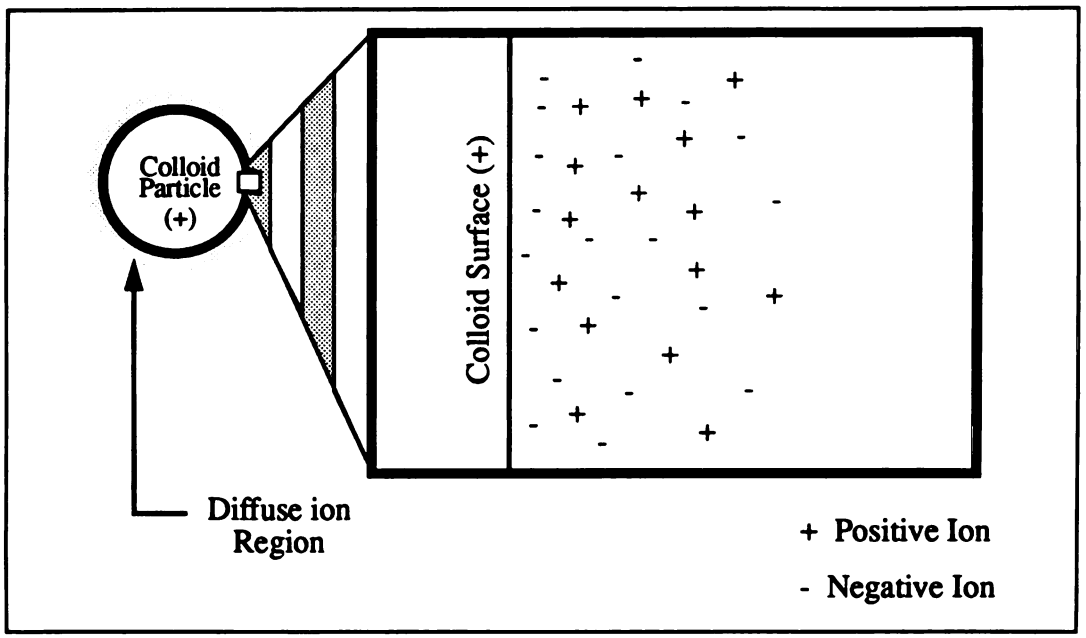


Figure 2.2 Gouy - Chapman Double Layer.

This figure shows an expanded view of the surface of the colloidal particle. The ions and coions migrate to the surface as shown.

where, q_i and q_j

and r is the dista

potential scalar f

potential, ψ_r is a

charge at that po

required to move

defines the electr

where, z is the va

Equation (2) desc

electric potential.

As outline

diffuse part of the

colloid particle re

electrical potential

where n_i are the co

potential energy of

of each ionic specie

Now that th

potential is ψ_r anoth

β at a point where t

that point, per unit v

where, q_i and q_j are two electric point charges, ε is the dielectric constant of the medium, and r is the distance between the charges. As a consequence of these forces, an electric potential scalar field exists between the charges. It is important to note that an electric potential, ψ, is associated with each point in space, whether or not there is any electric charge at that point. A change in the electric potential is equal to the amount of work required to move an electric charge from one point to another point. This latter statement defines the electrical potential energy and is defined as:

$$V_R = zq\psi, \tag{2}$$

where, z is the valence of the ion, q is the electronic charge and ψ is the electric potential. Equation (2) describes the general relation between the electrical potential energy and the electric potential.

As outlined by Shaw [9], the Gouy-Chapman description suggests that ions in the diffuse part of the double layer obey Boltzmann's distribution law. Boltzmann's law for a colloid particle relates the probability of ions being at a given point at which they have an electrical potential energy relative to the surface of the colloid, i.e.:

$$n_i = n_{i0}\exp\left(\frac{-z_i q\psi}{kT}\right) \tag{3}$$

where n_i are the concentrations of positive and negative ions at points where the electric potential energy of these ions are $zq\psi$ and $-zq\psi$, respectively, n_{i0} is the bulk concentration of each ionic species, z is the valence number of the ions and q is the electronic charge.

Now that the concentration of ions can be calculated at points where the electric potential is ψ, another definition, namely, the charge density, follows. The charge density, ρ, at a point where the electric potential is ψ, is defined as the sum of the charged ions, at that point, per unit volume. Mathematically, the charge density is:

Substituting equati

ρ

ρ

ρ

The next s

ic potential in o

ntities the charge

where ε is the di

potential and the

equation (8), the

The Poiss

for the electric po

description for w

part of the electric

tion to this equati

If, howeve

$$\rho = zqn_i \tag{4}$$

Substituting equation (3) into equation (4), then,

$$\rho = zq\left[n_{i0}\left(\exp\left[\frac{-zq\psi}{kT}\right] - \exp\left[\frac{zq\psi}{kT}\right]\right)\right] \tag{5}$$

$$\rho = -2zqn_{i0}\left[\frac{1}{2}\left(\exp\left[\frac{zq\psi}{kT}\right] - \exp\left[\frac{-zq\psi}{kT}\right]\right)\right] \tag{6}$$

$$\rho = -2zeqn_{i0}\sinh\left(\frac{zq\psi}{kT}\right) \quad . \tag{7}$$

The next step is to find another equation that relates the charge density with the electric potential in order to obtain an equation explicitly containing ψ. The Poisson equation relates the charge density to the Laplacian of the electric potential as:

$$\nabla^2\psi = \frac{-\rho}{\varepsilon}, \tag{8}$$

where ε is the dielectric constant of the medium, ρ is the charge density, ψ is the electric potential and the inverted triangle is the gradient operator. Substituting equation (7) into equation (8), the well known Poisson-Boltzmann equation is obtained:

$$\nabla^2\psi = \frac{2zqn_{i0}}{\varepsilon}\sinh\left(\frac{zq\psi}{kT}\right) \quad . \tag{9}$$

The Poisson-Boltzmann equation is a second order, non-linear differential equation for the electric potential. A solution to this equation will provide a quantitative scalar field description for which the magnitude of the electric potential at any location in the diffuse part of the electrical double layer can be calculated. Unfortunately, no exact analytical solution to this equation is known to exist and numerical methods must be used.

If, however, the assumption is made that the value of

n equation (9), t

Equation (11) fo

n zero. The ass

ne electric pote

Substituting the

equation simplif

where,

$$(\frac{zq\psi}{kT}) \ll 1 \tag{10}$$

in equation (9), then the following approximation holds:

$$\sinh(\frac{zq\psi}{kT}) \approx (\frac{zq\psi}{kT}) \quad . \tag{11}$$

Equation (11) follows from the Taylor expansion of sinh and setting the higher order terms to zero. The assumption in equation (10) implies that at room temperature, i.e. T = 25° C, the electric potential has the value of,

$$z\psi \ll \frac{kT}{e} = 25.69mV \, . \tag{12}$$

Substituting the right hand side of equation (11) into equation (9), the Poisson-Boltzmann equation simplifies to,

$$\nabla^2\psi = \kappa^2\psi \, , \tag{13}$$

where,

$$\kappa^2 = \frac{2z^2q^2n_{i0}N_A}{\varepsilon kT} \, , \tag{14}$$

ψ = the electric potential,
κ = the Debye - Hückel length,
z = valence number of ions,
q = the electronic charge,
q = 1.60217733 E-19 Coulomb,
n_{i0} = the bulk concentration of ions,
n_{i0} = n_{i0}(1000)(mole/meter^3),
N_A = Avagadros number,
N_A = 6.0221367 E+23 (1/mole),
ε = dielectric constant of the medium,
ε = $\varepsilon_0\varepsilon_r$,
ε_0 = permittivity of free space,
ε_0 = 8.8541878 E-12 (Farad/meter),
ε_r = the relative permittivity,
k = Boltzmann constant,

The assumption is

The Debye-Hücke

ion 12) $\psi < 25$

Hückel reciprocal

the layer. The thi

in which the elec

To unders

of radius, a, and

23 [8]. When a

is spread out.

For varia

Equation (15) is

be $\psi = \psi_0$ at $x =$

ure, the solutio

where, ψ_0 is th

the electric pote

A furthe

mentioned abov

forming a very

k = 1.380658 E-23 (Joules/Kelvin),
T = Temperature in Kelvins.

The assumption used to derive equation (13) is known as the Debye-Hückel approximation. The Debye-Hückel approximation is valid only for small electric potentials, i.e. from equation (12) $\psi < 25.69$ mV. The constant term κ in equation (13) is defined as the Debye-Hückel reciprocal length parameter and is an indicator of the thickness of the electrical double layer. The thickness of the double layer, $1/\kappa$, is the distance in the diffuse double layer in which the electric potential decays by a factor of 1/q for low potentials.

To understand the geometry of the double layer, consider a spherical colloid particle of radius, a, and the ratio of this particle radius to the double layer thickness, $a\kappa$, see Figure 2.3 [8]. When $a\kappa$ is large, the double layer is nearly flat. When $a\kappa$ is small, the double layer is spread out.

For variations in the potential in the x-direction, equation (13) takes the form,

$$\frac{\partial^2 \psi}{\partial x^2} = \kappa^2 \psi \ . \tag{15}$$

Equation (15) is a second order linear differential equation. Letting the boundary conditions be $\psi = \psi_0$ at x = 0 and $\psi = 0$ at x = infinity, and assuming low potentials at room temperature, the solution to equation (15) is,

$$\psi = \psi_o \exp(-\kappa x) \ , \tag{16}$$

where, ψ_0 is the electric potential at the surface of the particle. Equation (16) shows that the electric potential decays exponentially from the surface of the colloid.

A further attribute of the electrical double layer was introduced by Otto Stern. As mentioned above, thermal agitation prevents counterions in the suspension medium from forming a very compact layer surrounding the charged colloid. If, however, the electrostatic

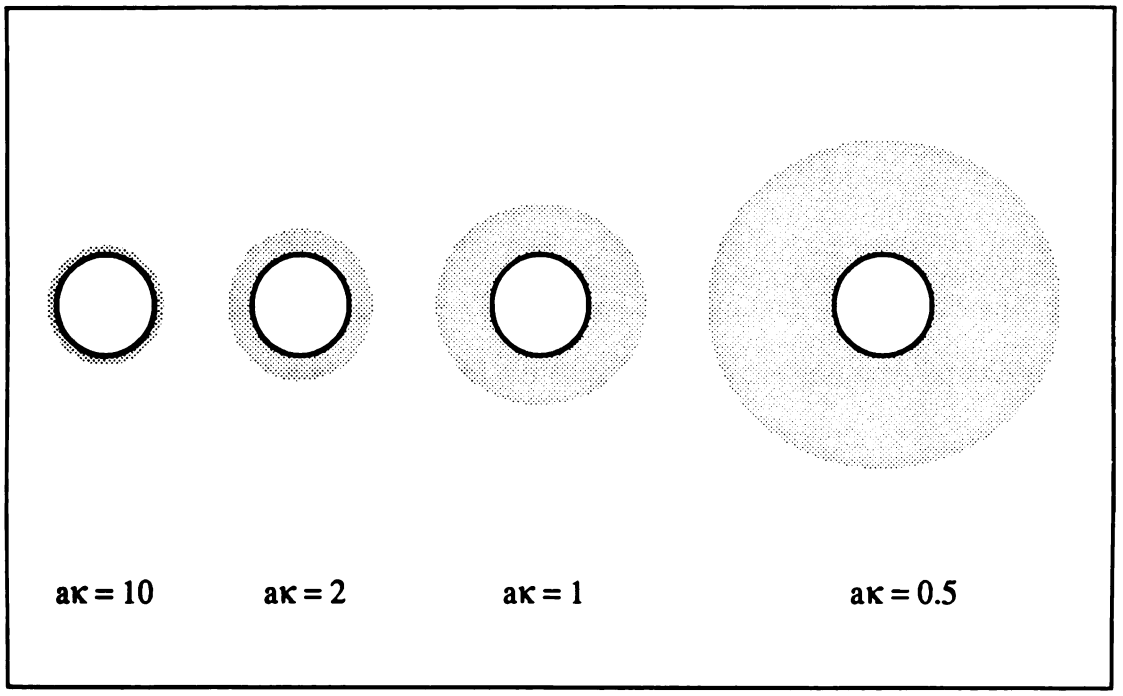


Figure 2.3 Ratio of the particle radius to the double layer thickness.

The magnitudes of aκ vary in this figure. Most ceramic colloids have magnitudes of aκ in the proximity of 50 to 100.
Mysels, Karol J., Introduction to Colloid Chemistry, Robert E. Krieger Publishing Company, Huntington, New York, 1978.

forces are too st
overcome them.
called the Stern
The new
shown in Figure
The elect
ata potential. E
at the Stern laye
determined usin
measured by ele
ating electric fi
induces a force
velocity. A lase
velocities.
The elec
magnitude of th
potential by the
Where, ζ is the
the dielectric co
known as the H
The acou
sures the zeta p
suspension. Aga

forces are too strong near the colloids surface, then thermal agitation will not be able to overcome them. The result is a semi-compact layer of counterions surrounding the colloid called the Stern layer.

The new picture of the electrical double layer in terms of the electrical potential is shown in Figure 2.4 [8].

The electric potential at the plane of shear near the Stern potential is defined as the zeta potential. Exactly how close the zeta potential lies in relation to the electric potential at the Stern layer is a topic of current research. The zeta potential can be experimentally determined using various techniques such as micro-electrophoresis or acoustophoresis as measured by electrokinetic sonic amplitude (ESA). Micro-electrophoresis applies an oscillating electric field to a colloid suspension. The presence of the external electric field induces a force that acts on each colloid causing the colloid particles to move with a certain velocity. A laser beam and a detector are used to optically measure the colloid particles velocities.

The electrophoretic mobility is determined by dividing the observed velocity by the magnitude of the electric field. The electrophoretic mobility is directly related to the zeta potential by the following equations [2]:

$$\varsigma = \frac{6\pi\eta\mu}{\varepsilon} \quad \text{when } a\kappa < 0.1 \tag{17}$$

$$\varsigma = \frac{4\pi\eta\mu}{\varepsilon} \quad \text{when } a\kappa > 100. \tag{18}$$

Where, ζ is the zeta potential, η is the viscosity, μ is the electrophoretic mobility and ε is the dielectric constant. Equation (17) is known as the Hückel equation and equation (18) is known as the Helmholtz - Smoluchowski equation.

The acoustophoresis technique using an electrokinetic sonic amplitude (ESA) measures the zeta potential by applying a one megahertz oscillating electric field to a colloid suspension. Again, the particles move with a certain velocity due to the effects of the elec-

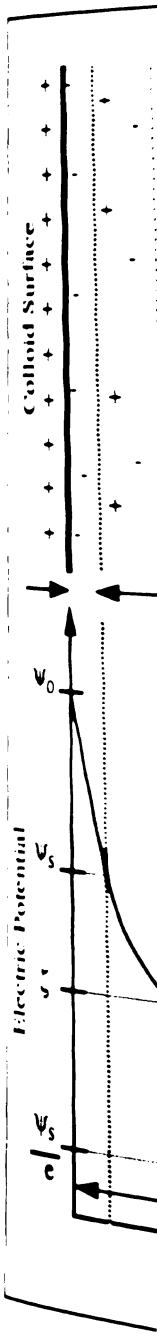


Figure 2.4 The

This figure show
line. The ions an
shape of decay o
Mysels, Karol J
Company, Hunti

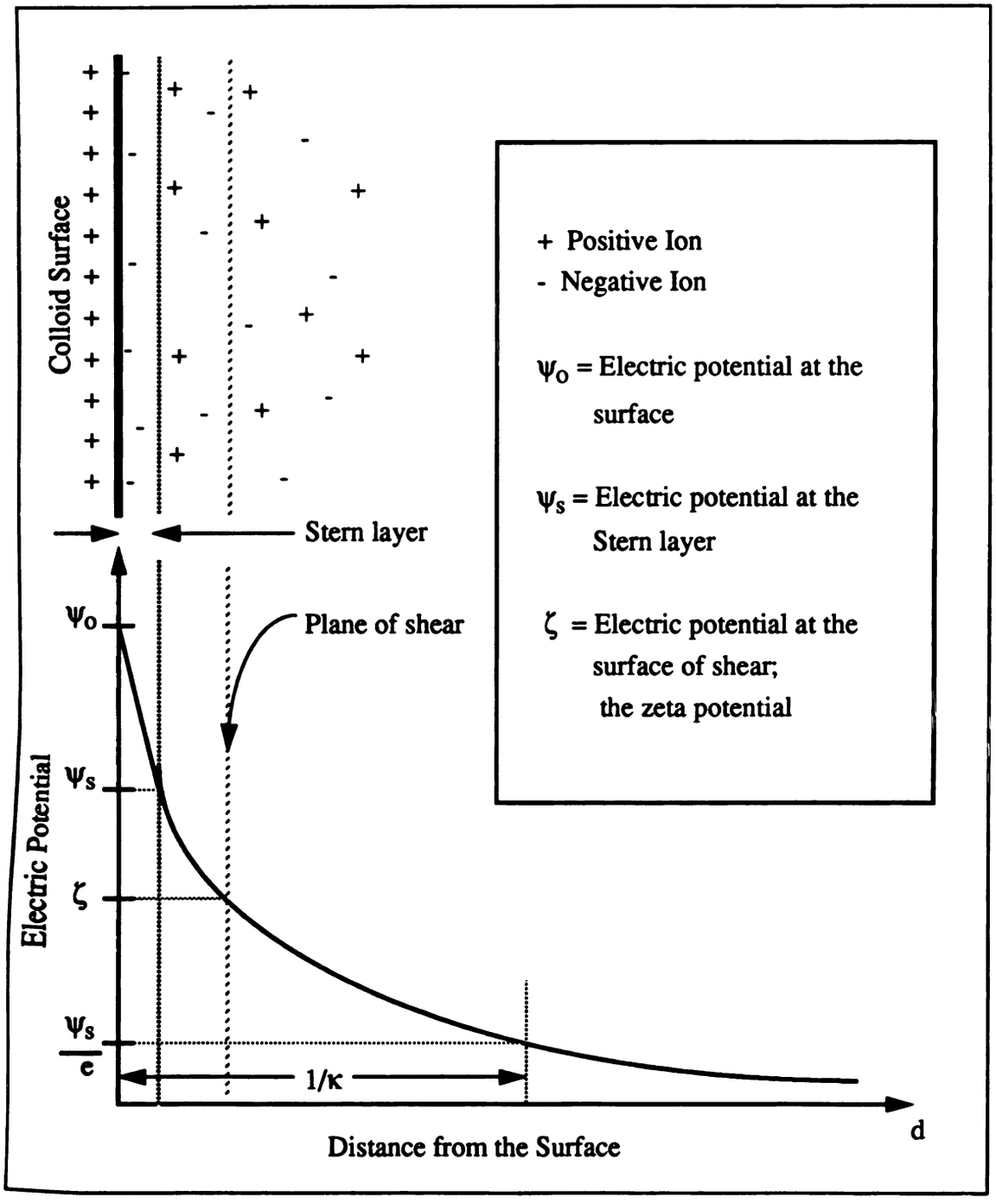


Figure 2.4 The Stern electrical double layer.

This figure shows the small scale surface of a colloid particle, represented as a vertical line. The ions and coions are depicted as + and - symbols. The lower graph shows the shape of decay of the electric potential moving away from the surface of the particle. Mysels, Karol J., Introduction to Colloid Chemistry, Robert E. Krieger Publishing Company, Huntington, New York, 1978.

tric field. As t

pressure wav

sonic detecto

may be deter

zeta potentia

When

result is an el

energy betwe

the potential

of concentric

(HHF) [12]. S

trical potentia

tion energy be

$$V_R = \frac{\varepsilon a_1 a_2}{4}$$

where, ε is the

one and partic

the Debye-Hü

cle's surfaces.

ified, equation

independent va

surface potent

potential energ

Levine and Mc

The nex

tric field. As the colloid particles vibrate back and forth in the suspension medium, a sonic pressure wave is produced. The frequency of the sound wave is measured by a sensitive sonic detector. Once the frequency of the sound wave is known, the velocity of the particle may be determined as well as the electrophoretic mobility. Using equations (17) or (18), the zeta potential is then obtained.

When the diffuse regions of two double layers surrounding two colloids overlap, the result is an electrical potential energy of interaction. Derjaguin [10] derived the potential energy between two parallel plates of unequal charge. Using the calculations of Derjaguin, the potential energy between two unequal spheres was determined using a summation idea of concentric, parallel plates. This method was employed by Hogg, Healy and Fuerstenau (HHF) [12]. Starting with the electric potential and relating the electric potential to the electrical potential energy, Hogg, Healy and Fuerstenau [12] obtained the electrostatic interaction energy between two dissimilar spherical particles,

$$V_R = \frac{\varepsilon a_1 a_2 (\psi_{01}^2 + \psi_{02}^2)}{4 (a_1 + a_2)} \left[\left(\frac{2\psi_{01}\psi_{02}}{\psi_{01}^2 + \psi_{02}^2} \right) \ln \left(\frac{1 + \exp(-\kappa H_0)}{1 - \exp(-\kappa H_0)} \right) + \ln (1 - \exp(-2\kappa H_0)) \right] \tag{19}$$

where, ε is the dielectric constant of the medium, a_1, a_2 are the respective radii of particle one and particle two, ψ_{01}, ψ_{02} are the electric potentials at the surface of each particle, κ is the Debye-Hückel reciprocal length parameter and H_0 is distance between the two particle's surfaces. When the initial conditions for two spherical colloids in suspension are specified, equation (19) has the displacement between the two particles surfaces as its only independent variable. HHF [11] also show that equation (19) is a valid approximation for surface potentials less than approximately 50 to60 mV. Other methods for obtaining the potential energy relation between two dissimilar spherical particles are given by Bell, Levine and McCartney [12] and by Bell and Peterson [13].

The next topic to consider is the London - van der Waals attraction force between

colloid particles.

by Mahanty and

The conce

larger than molec

Waals developed

directly related t

between two dip

tion energy of a

two molecules.

developed by Lo

due to the Londo

where, A is the

cules. The calcu

icles has been

molar model of

is based on pai

colloid particle

formulas and t

Hamaker mode

Integra

the potential en

given by Ham

colloid particles. A brief summary of the evolution of the nature of this force is described by Mahanty and Ninham [14] and is outlined below.

The concept of a force field existing between any pair of molecules whose range is larger than molecular dimensions was first investigated by van der Waals in 1873. van der Waals developed an equation of state for a gas in which a constant term appeared that was directly related to the strength of the intermolecular forces. By averaging the interaction between two dipoles over all orientations, van der Waal and others found that the interaction energy of a dipolar molecule was proportional to $1/r^6$, where r is the distance between two molecules. The explanation of the force between a pair of non-polar molecules was developed by London in 1930. The attractive interaction energy between two molecules due to the London - van der Waal force was determined to be:

$$E(r) = -\frac{\Lambda}{r^6} , \tag{20}$$

where, Λ is the London - van der Waals constant and r is the distance between the molecules. The calculation of the electrodynamic attraction force between two macroscopic particles has been approached by two different methods. The Lifshitz model is based on a molar model of condensed media and uses quantum electrodynamics. The Hamaker model is based on pairwise summation of the attractive energies between the molecules of each colloid particle, ignoring multibody perturbations. Due to the complexity of the Lifshitz formulas and the necessity for numerical methods for determining material functions, the Hamaker model will be considered.

Integration of equation (20) over the total volumes of two colloid particles provides the potential energy of interaction between two particles containing q atoms per cm^3 and is given by Hamaker [15] as:

$$V_A = -\int_{V_1} dv_1 \int_{V_2} \frac{q^2 \Lambda}{r^6} dv_2 , \tag{21}$$

where. dv_1, dv_2. V

spectively. r is th

onstant. Hamaker

pherical particles.

Consider a sphere

Figure 2.5).

The sphere

around point P wi

where. θ_0 is given

Integrating equatio

The volume eleme

The potential ener

E

where, dv_1, dv_2, V_1 and V_2 are volume elements and total volumes of the two particles respectively, r is the distance between dv_1 and dv_2 and Λ is the London - van der Waals constant. Hamaker derived an equation for the attractive interaction energy between two spherical particles. His derivation begins by investigating a molecule near a sphere. Consider a sphere of radius a_1 with center at point O and a point P at a distance $\overline{OP} = R > a_1$. (Figure 2.5).

The sphere around O cuts out a surface, S(ABC) out of a second sphere centered around point P with radius r. The surface S(ABC) is:

$$S(ABC) = 2\pi \int_0^{\theta_o} r^2 \sin\theta d\theta \ . \tag{22}$$

where, θ_0 is given by the law of cosines:

$$a_1 = R^2 + r^2 + 2rR\cos\theta_o \ . \tag{23}$$

Integrating equation (22), the surface ABC is:

$$S(ABC) = \pi \frac{r}{R} (a_1^2 - (R-r)^2) \ . \tag{24}$$

The volume element dv_1 is given by:

$$dv_1 = S(ABC)\, dr \ . \tag{25}$$

The potential energy of a molecule at P may then be written as:

$$E_P = - \int_{(R-a_1)}^{(R+a_1)} \frac{\Lambda q}{r^6} \left(\frac{\pi r}{R}\right) (a_1^2 - (R-r)^2)\, dr. \tag{26}$$

Figure 2.5 N

This figure de
point O.
Hamaker, H. C
tion Between S

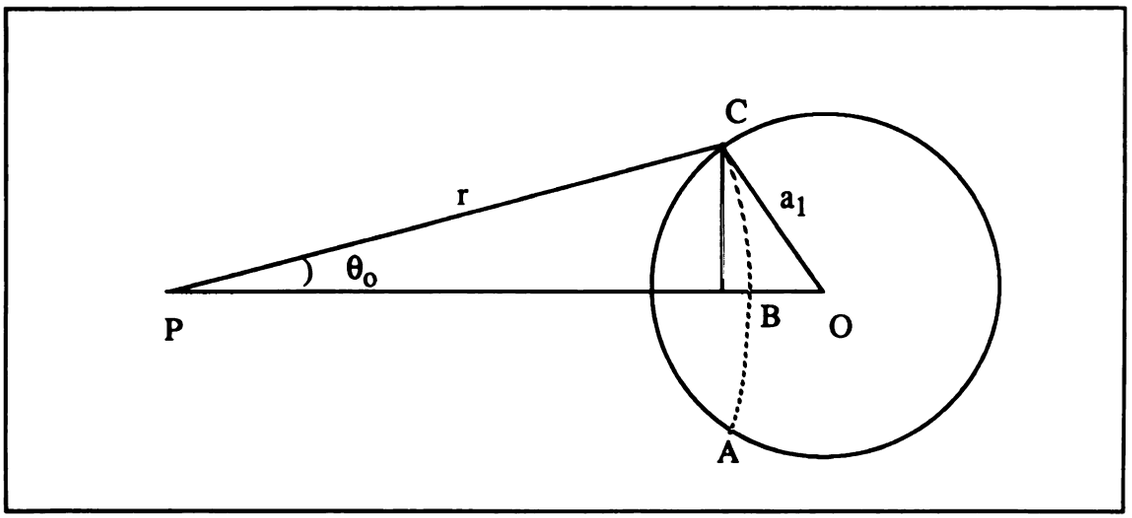


Figure 2.5 Molecule near a sphere.

This figure depicts a molecule at point P at a distant OP from a sphere centered at point O.
Hamaker, H. C., *Physica* 4, 1058 - 1072, 1937, "The London - van der Waals Attraction Between Spherical Particles".

The potential ener

s, with centers be

The result of thi

yields.

$$V_A = -\frac{A}{6}$$

where A is the H

spheres.

The poten

ing one of the sp

where,

A is the Hamak

face of the plane

The deve

the London - va

interaction step

involve the forc

dynamic forces.

The potential energy of interaction between two spheres, the second sphere having radius a_2, with centers being a distance C apart is obtained from the following:

$$V_A = \int_{(C-a_2)}^{(C+a_2)} E_P q \pi \frac{R}{C} (a_2^2 - (C-R)^2)\, dR. \tag{27}$$

The result of this integration for the potential energy of interaction between two spheres yields,

$$V_A = -\frac{A}{6}\left(\frac{2a_1a_2}{C^2-(a_1+a_2)^2} + \frac{2a_1a_2}{C^2-(a_1-a_2)^2} + ln\left(\frac{C^2-(a_1+a_2)^2}{C^2-(a_1-a_2)^2}\right)\right), \tag{28}$$

where A is the Hamaker constant and C is the distance between the centers of the two spheres.

The potential energy of interaction for a sphere and a plane can be calculated by letting one of the sphere's radius go to infinity and has the form [14]:

$$V_A = -\frac{A}{6}\left(\frac{1}{x} + \frac{1}{2+x} + ln\left(\frac{x}{2+x}\right)\right), \tag{29}$$

where,

$$x = \frac{(C-a_1)}{a_1}. \tag{30}$$

A is the Hamaker constant, C is the distance between the center of the sphere and the surface of the plane, and a_1 is the radius of the sphere.

The development of the potential energy equations due to both the double layer and the London - van der Waals force will be used to determine the dynamics of the surface interaction step of the electrophoretic deposition process. The transport step dynamics involve the forces listed in table 2.2 that may be present in the system, excluding electrodynamic forces. The dynamical equations of motion and the numerical algrithm that may

implemented c

be implemented on a computer will be derived in the following chapter.

To examin

specified in order t

sion will clarify w

will be used to de

Because th

namic forces that

be included. Alth

negligible, for an

combined mass o

attractive force w

of two electrical

For the F

the repulsive ste

Finally, external

obtained from th

Other ini

lowing:

1. T
b
2. T
s
3. E
4. T
p
5. T
a

FORMULATION OF EQUATIONS

To examine the dynamics of a colloidal suspension, initial conditions need to be specified in order to isolate what forces will be present in the system. The following discussion will clarify what forces will be used for the derivation of the equations of motion that will be used to describe the dynamics of the colloidal system.

Because the suspended particles will be moving in a viscous medium, hydrodynamic forces that obey Stoke's law will be included in the system. Gravitational forces will be included. Although the magnitude of the gravitational force on an individual colloid is negligible, for an unstable system in which the colloids flocculate to form agglomerates, the combined mass of several particles can lead to sedimentation. The London - van der Waal attractive force will be included. The electrostatic repulsive force arising from the overlap of two electrical double layers will be included as well.

For the FED process, no polymer chains will be added to the system and therefore the repulsive steric force will not be included. Brownian motion will not be included. Finally, external magnetic or electric fields will not be present and so the resulting forces obtained from these fields will not be included.

Other initial conditions that will be imposed on the colloidal system include the following:

1. The colloidal particles will be assumed to be spherical in shape and insoluble.
2. The surface of each particle will be assumed to have a constant charge density.
3. Each spherical particle will be assumed to be infinitely hard and smooth.
4. The zeta potential will be used as the numerical equivalent of the surface potential in calculations.
5. The frame of reference used to specify the particles coordinates will be assumed to be an inertial frame of reference.

The equation

medium is given

where, m is the

F_{Vm} is the Lon

drag and F_g is t

The force due t

where, r is the

densities and g

where, η is the

velocity.

The electrodyn

conserved syst

where F is the

dient operator.

ing electrodyna

The equation of motion for an individual spherical, colloidal particle suspended in a medium is given by Newton's second law of motion,

$$m\frac{d\vec{v}}{dt} = \vec{F}_{DBL} + \vec{F}_{Van} + \vec{F}_{HYD} + \vec{F}_g = \vec{F}_{TTL} \,, \tag{31}$$

where, m is the mass of the colloid particle, F_{DBL} is the Coulombic, double layer force, F_{Van} is the London - van der Waals force, F_{HYD} is the hydrodynamic force due to Stoke's drag and F_g is the force due to gravity.

The force due to gravity for a sphere in a medium is given as:

$$\vec{F}_g = -\frac{4}{3}\pi r^3 (\rho_2 - \rho_1)\,\hat{g} \,, \tag{32}$$

where, r is the particle radius, ρ_2, ρ_1 are the respective particle and suspension medium densities and g is the acceleration due to gravity. The Stoke's drag force for a sphere is:

$$\vec{F}_{HYD} = 6\pi\eta r\vec{v} \,, \tag{33}$$

where, η is the suspension medium viscosity, r is the particle radius and v is the particle's velocity.

The electrodynamic forces can be obtained from the following relation that is valid for a conserved system:

$$\vec{F} = -\nabla \vec{U} \,, \tag{34}$$

where F is the total force, U is the potential energy equation and the del symbol is the gradient operator. Applying equation (34) to the potential equations, (19) and (28) the following electrodynamic force equations are:

The repulsive fo

$F_{rel}(PP)$ =

where,

The attractive

where,

The system c

where N is a

The n

ticle as time

known. A ge

The repulsive force between two non-identical, spherical particles is:

$$\hat{F}_{DBL}\ [PP] = -S_1\left(\frac{2\varsigma_1\varsigma_2\left(-\frac{\kappa exp\,(-\kappa H)}{1-exp\,(-\kappa H)}-\frac{(1+exp\,(-\kappa H))\,kexp\,(-\kappa H)}{(1-exp\,(-\kappa H))^2}\right)(1-exp\,(-\kappa H))}{(\varsigma_1^2+\varsigma_2^2)\,(1+exp\,(-\kappa H))}+S_2\right) \quad (35)$$

where,

$$S_1 = \frac{\varepsilon r_1 r_2\,(\varsigma_1^2+\varsigma_2^2)}{4r_1+4r_2} \quad (36)$$

$$S_2 = \frac{2\kappa exp\,(-2\kappa H)}{1-exp\,(-2\kappa H)}\ . \quad (37)$$

The attractive force between two non-identical, spherical particles is:

$$\hat{F}_{Van}[PP] = \frac{A}{12}\left(-\frac{r_2T_2}{r_1T_3}-\frac{r_2T_2}{r_1T_1^2}+\frac{2T_1}{T_3}\left(\frac{T_2}{T_1}-\frac{T_3T_2}{T_1^2}\right)\right) \quad (38)$$

where,

$$T_1 = \frac{H^2}{4r_1^2}+\frac{Hr_2}{2r_1^2}+\frac{H}{2r_1}+\frac{r_2}{r_1} \quad (39)$$

$$T_2 = \frac{H}{2r_1^2}+\frac{r_2}{2r_1^2}+\frac{1}{2r_1} \quad (40)$$

$$T_3 = \frac{H^2}{4r_1^2}+\frac{Hr_2}{r_1^2}+\frac{H}{2r_1} \quad (41)$$

and H is the distance between the two particles surfaces.

The system of n colloidal particles produces N equations of motion, one for each particle, where N is an integer.

The many body problem involves determining the position and velocity of each particle as time progresses, provided that the initial position and velocity of each particle are known. A general solution to the many body problem for N greater than three is unknown

and numerical

the basic proces

vide the necessa

bital system.

Many body prob

1. Define t

A. Part

1 C
2 C
3 S
4 I
5 I
6 I
7
8 I

B. Med

1 C
2 C
3 I
4
5 I
6
7 I

C. Elec

1 C
2 C
3 C

D. Co

1 C
2 C
3 S
4 I
5 I
6 I

and numerical methods must be employed. The flow chart presented in Figure 3.1 shows the basic process used to solve the many body problem. The following algorithm will provide the necessary steps that outline a solution to the many body problem applied to a colloidal system.

Many body problem Algorithm:

1. Define the initial values of the colloidal system.

A. Particle information	Units
1 Common Name	-
2 Chemical Name	-
3 Shape	-
4 Density	gm/cm^3
5 Diameter mean	µm
6 Diameter Standard Deviation	µm
7 Number of Particles	-
8 Hamaker Constant	Joules

B. Medium Information	Units
1 Common Name	-
2 Chemical Name	-
3 Density	gm/cm^3
4 Viscosity	gm/(cm sec)
5 Relative permittivity	-
6 Temperature	Degrees Celsius
7 Hamaker Constant	Joules

C. Electrolyte Information	Units
1 Common Name	-
2 Chemical Name	-
3 Concentration	Normality

D. Collector Information	Units
1 Common Name	-
2 Chemical Name	-
3 Shape	-
4 Density	gm/cm^3
5 Diameter (Aspect Ratio)	µm
6 Hamaker Constant	Joules

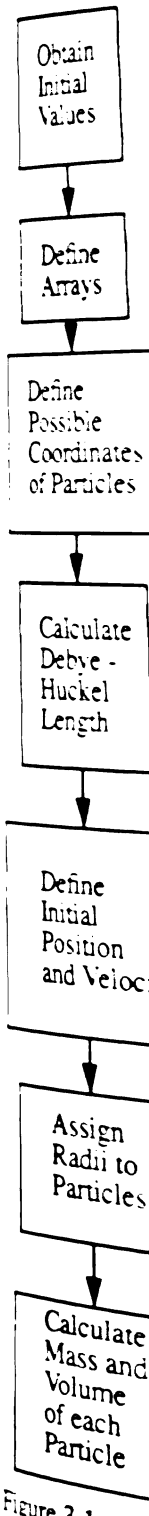


Figure 3.1 F

This figure be
required to sol

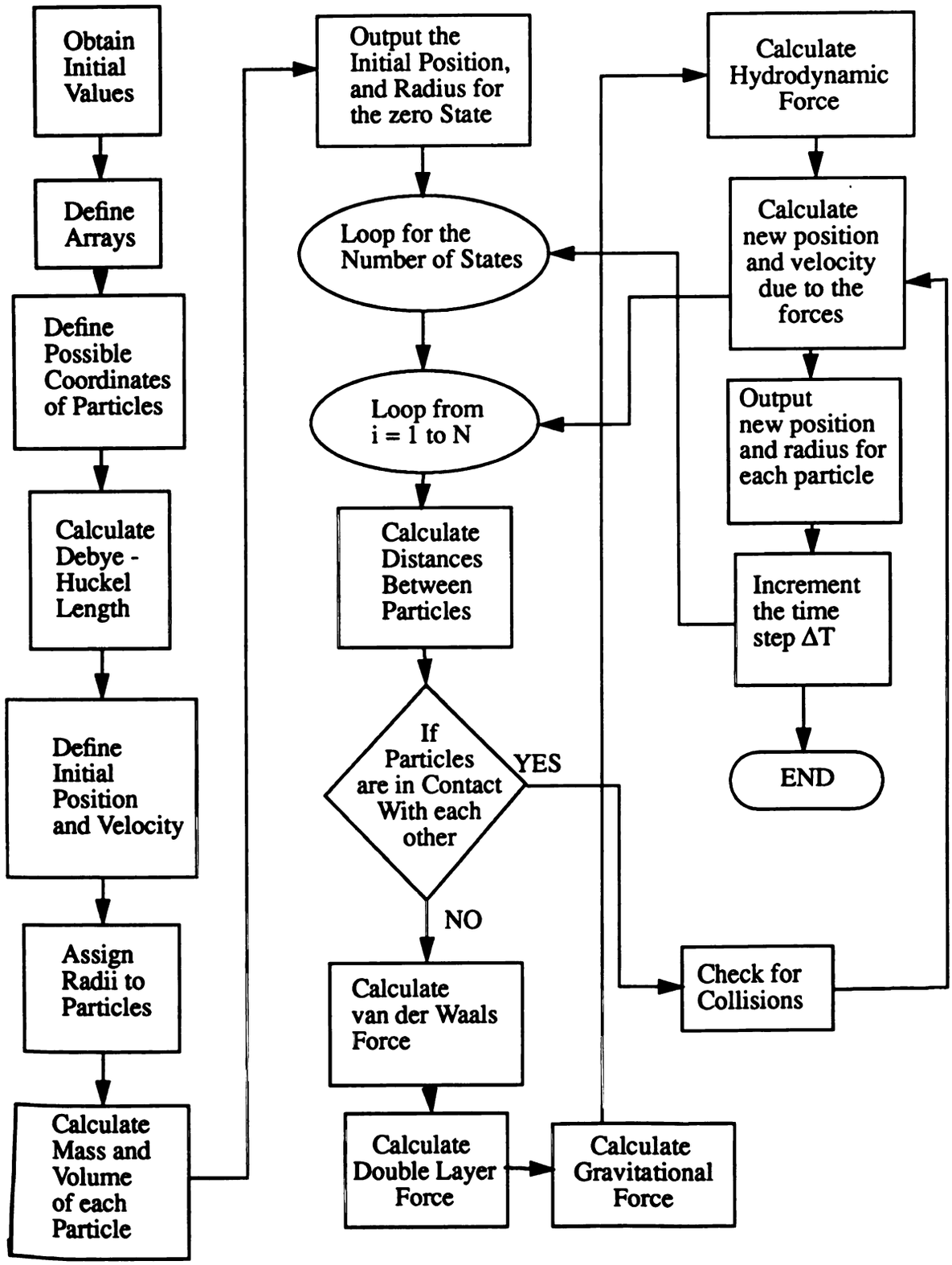


Figure 3.1 Flow chart for solving the many body problem.

This figure begins in the upper left corner and progresses through the necessary steps required to solve multiple colloidal bodies interacting with each other.

E. S

1 P
2 A
3 A

F. Tim

1 I

G. Dyn

1 I
2
3

H. Sim

1
2

All

number

ticle de

1.H

suspens

to equa

a spher

betwee

particle

1.C

Verwey

loidal

regardl

individ

three m

E. Surface Parameters	Units
1 pH Level	-
2 Zeta potential of the particle	mV
3 Zeta potential of the Collector	mV

F. Time Increment between Calculations	Units
1 ΔT	seconds

G. Dynamic Model

1 DLVO
2 Acid / Base
3 Random

H. Simulation Type

1 Particle A - Collector
2 Particle A - Particle B

All of the initial values will be referenced throughout this algorithm by the step number proceeded by a letter proceeded by an index number. For example, the particle density will be referenced by 1.A.4.

1.H describes a simulation type. Simulation 1.H.1 involves colloidal particles in suspension interacting with a plane-shaped collector. Equation (34) may be applied to equation (29) to obtain a force equation for the van der Waals attraction between a sphere and a plate. A similar formula may be obtained for the double layer force between a sphere and a plane. Simulation 1.H.2 involves the interaction of colloidal particles of type A collecting onto particles of type B.

1.G are the possible dynamic models available. 1.G.1 is the Derjaguin, Landau, Verwey and Overbeek model. 1.G.2 is an acid/base model that claims that when colloidal particles come into close proximity to one another the particles coagulate regardless of what forces may be present. 1.G.3 is a random model that allows the individual colloids to sample from a time dependent force distribution. Each of the three models affects the magnitude of the electrodynamic forces.

2. Defin

ticle.

Let N

Array

Dens

Mass

Radiu

Volun

x_{xi}

x_{yi}

v_{xi}

v_{yi}

dx/dt

dy/dt

dv_x/dt

dv_y/dt

Distance

F_{TTL}

F_{xi}

F_{xi}

3. Define the

cles will be

2. Define the necessary arrays to hold the pertinent information for each colloidal particle.

Let N be the number of particles obtained from (1.A.7).

Array Name	Dimension	Comment
Density	N x 1	Particle Density
Mass	N x 1	Particle Mass
Radius	N x 1	Particle Radius
Volume	N x 1	Particle Volume
x_{xi}	N x 1	x Position
x_{yi}	N x 1	y Position
v_{xi}	N x 1	x Velocity
v_{yi}	N x 1	y Velocity
dx/dt	4 x N	The four previous values of v_{xi}
dy/dt	4 x N	The four previous values of v_{yi}
dv_x/dt	4 x N	The four previous values of acceleration.
dv_y/dt	4 x N	The four previous values of acceleration.
Distance	N x N	The distance between particle i and particle j
F_{TTL}	N x N	The total force acting on particle i due to particle j
F_{xi}	N x 1	The x component of F_{TTL}
F_{xi}	N x 1	The x component of F_{TTL}

3. Define the display dimensions and the possible coordinates where the initial particles will be placed.

The display size may be sized as desired but for this thesis the following values will be used:

Display Dimensions:	150 units x 100 units
Maximum Number of Particles:	120
Maximum Particle Diameter:	2 units

Define an 8 row by 15 column grid centered in the middle of the display. Let each row and each column be separated by 3 units. The 8 row by 15 column grid defines 120 boxes, each box having dimension 3 units by 3 units.

Figure 3.2 shows the graphic display that will be used to visualize the suspended colloidal particles.

To prevent the particles from overlapping one another, each particle is initially positioned at the center of an unoccupied box in the grid.

4. Calculate the Debye-Hückel length using equation (14):

$$\kappa^2 = \frac{2z^2q^2n_0N_A}{\varepsilon kT} . \tag{42}$$

5. Define the initial position and velocity of each particle at time $t = t_0$.

$x_{xi}(t_0) = x_{xi}^o$,

$x_{yi}(t_0) = x_{yi}^o$,

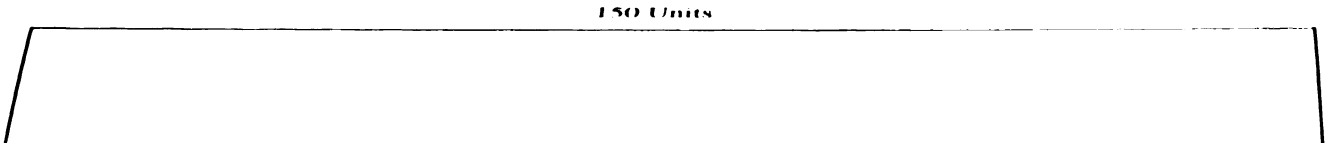
$v_{xi}(t_0) = v_{xi}^o$,

$v_{yi}(t_0) = v_{yi}^o$.

6. Assign a radius to each particle assuming that the particle diameters obey a Gaussian distribution.

To calculate a gaussian random number from two uniform random numbers do the following:

Obtain the Diameter mean, μ, from 1.A.5 and obtain the Diameter standard devia-


150 Units

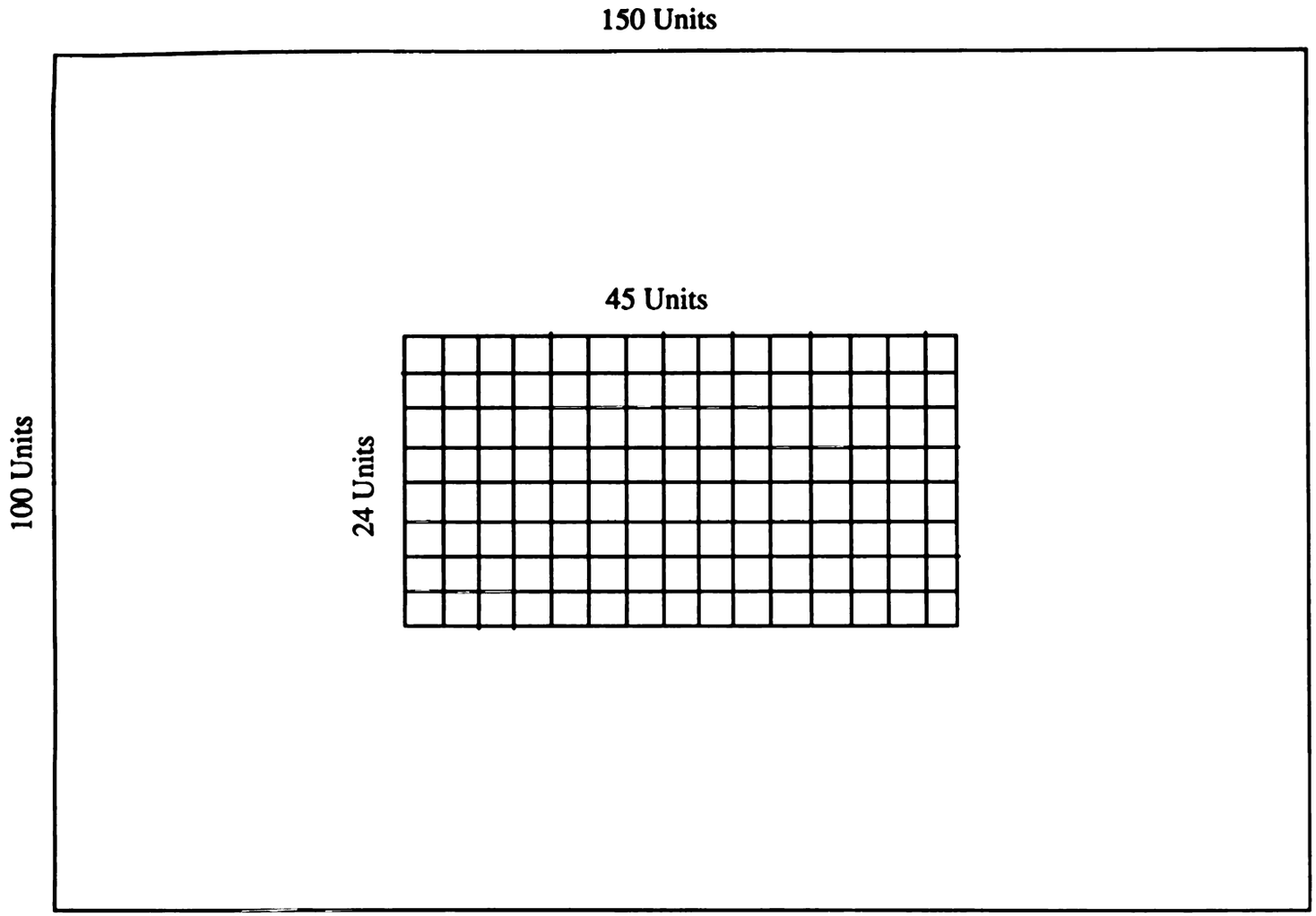


Figure 3.2 Graphic display used to visualize the suspended colloids.

This figure shows the display screen for the simulation and the x and y dimensions.The grid shown is not actually displayed. Each particle is centered in one of the boxes to prevent initial overlap of particles.

tion, σ, from 1.A.6. Let u_1 and u_2 be two uniform random numbers whose values range from 0 to 1. Define the parameter G as:

$$G = \sqrt{-2 \times \log u_1} \times \cos(2\pi u_2) \ . \tag{43}$$

Assign the radius to be:

$$r_i = \mu + \sigma G. \tag{44}$$

7. Calculate the Volume and Mass of each particle.

$$V_i = \frac{4}{3}\pi r_i^3 \ , \tag{45}$$

$$m_i = \rho_i V_i \ , \tag{46}$$

where, V_i is the volume of the i-th particle using the radius from equation (44), m_i is the mass of the i-th particle, ρ_i is the density of the i-th particle obtained from 1.A.4 and i is an integer, i.e. i= 1, 2, ..., N.

8. Assign the initial value conditions for each particle.

$(dx/dt)_{0,i} = v_{xi}$, $(dv_x/dt)_{0,i} = 0$,

$(dy/dt)_{0,i} = v_{yi}$, $(dv_y/dt)_{0,i} = 0$

9. Output the initial x, y positions and the corresponding radii.

10. Calculate the state of the colloidal system at each time increment. The state of the system is determined by calculating the position and velocity of each particle at a specified time. The time between states is incremented by Δt obtained from 1.F.

Loop until the number of states have been generated. (Each loop through the following sub-algorithm will generate a state of the system.) Generally, 30 states are needed for one second of animation.

Let j = 1

Loop from i = 1 to N

If j <> i then calculate the total distance between particles i and j.

$$(D)_i = \sqrt{(x_{xj} - x_{xi})^2 + (x_{yj} - x_{yi})^2} \tag{47}$$

If the distance between particle i and particle j is less than δ units then

Calculate the relative Hamaker constant

$$A_{123} = (A_{11}^{0.5} - A_{22}^{0.5})\,(A_{33}^{0.5} - A_{22}^{0.5}) \quad (48)$$

A_{123} is the relative Hamaker constant of substance 1 and 3 that are separated by substance 2. A_{11} is the Hamaker constant for particle A, A_{33} is the Hamaker constant for particle B and A_{22} is the Hamaker constant for the medium.

Calculate the London van der Waals force between particle i and particle j using equations (38), (39), (40) and (41).

If the distance between particle i and the collector is less than δ units then

Calculate the London van der Waals force between particle i and the collector.

Calculate the dielectric constant of the medium using 1.B.5, the medium permittivity and the following equation:

Dielectric Constant = (1.B.5) x (8.854187799 x 10^{-12}.

Use the zeta potential data from (1.E.2) and (1.E.3).

Calculate the double layer repulsive force between particle i and particle j using equations (35), (36) and (37).

If the distance between particle i and the collector is less than δ units then

Calculate the double layer repulsive force between particle i and the collector.

Calculate the total force acting on particle i due to particle j.

$$\vec{F}_{TTL} = \vec{F}_{Van} + \vec{F}_{DBL} \quad (49)$$

Calculate the x and y force components on particle i:

$$F_{xi} = F_{TTL_i} \times \frac{(x_{xj} - x_{xi})}{(D)_{ij}}, \quad (50)$$

where D is the distance between the particles.

$$F_{yi} = F_{TTL_i} \times \frac{(x_{yj} - x_{yi})}{(D)_{ij}}. \tag{51}$$

Else if the distance between particle i and particle j is equal to $r_i + r_j$ then

Use the following elastic collision algorithm:

For a perfectly elastic collision, the coefficient of restitution, e = 1. The magnitudes of the velocity are known from v_{xi} and v_{yi}. The direction of the velocity may be obtained from x_{xi}, x_{xi-1}, x_{yi} and x_{yi-1}. Define a vector, n, that is normal to the centers of the two particles. Define a tangent vector, t, that is perpendicular to n. Resolve the velocity into components along the t and n vectors.The impulse forces are directed along the vector n. The t components of velocity are unchanged after the collision. Use the following two equations to determine the new direction and magnitudes of the velocity after the collision [16]:

$$m_i (v_i)_n + m_j (v_j)_n = m_i (v_i^p)_n + m_j (v_j^p)_n \tag{52}$$

$$(v_j^p)_n - (v_i^p)_n = e\,((v_j)_n - (v_i)_n) \tag{53}$$

Add the force due to gravity using equation (32). (Note that the force of gravity acts only on the y component.)

$$\vec{F}_{yi} = \vec{F}_{yi-1} - (-\frac{4}{3}\pi r^3 (\rho_2 - \rho_1)\,\vec{g}) \tag{54}$$

where ρ_2 is the particle density from (1.A.4) and ρ_1 is the medium density from (1.B.3).

Add the Hydrodynamic force obeying Stoke's law using equation (33).

$$\vec{F}_{xi} = \vec{F}_{xi-1} - (6\pi\eta r_i \vec{v}_{xi-1}) \tag{55}$$

$$\vec{F}_{yi} = \vec{F}_{yi-1} - (6\pi\eta r_i \vec{v}_{yi-1}) \tag{56}$$

Use the Euler modified method to find the first three values of position and velocity.

$$x_{xi} = x_{xi-1} + \Delta t\left(v_{xi-1} + \frac{\Delta t \times F_{xi}}{2m_i}\right) \tag{57}$$

$$x_{yi} = x_{yi-1} + \Delta t\left(v_{yi-1} + \frac{\Delta t \times F_{yi}}{2m_i}\right) \tag{58}$$

$$v_{xi} = v_{xi-1} + \Delta t\left(\frac{F_{xi}}{m_i}\right) \tag{59}$$

$$v_{yi} = v_{yi-1} + \Delta t\left(\frac{F_{yi}}{m_i}\right) \tag{60}$$

$$\left(\frac{dx}{dt}\right)_{k,i} = v_{xi} \tag{61}$$

$$\left(\frac{dy}{dt}\right)_{k,i} = v_{yi} \tag{62}$$

$$\left(\frac{dv_x}{dt}\right)_{k,i} = \left(\frac{F_{xi}}{m_i}\right) \tag{63}$$

$$\left(\frac{dv_y}{dt}\right)_{k,i} = \left(\frac{F_{yi}}{m_i}\right) \tag{64}$$

where k = 1, 2, 3.

Use the fourth order Adams - Bashforth method to find the new values of position and velocity for the remaining particles.

$$x_{xi} = x_{xi-1} + \frac{\Delta t}{2.4}\left(5.5\left(\frac{dx}{dt}\right)_{3,i} - 5.9\left(\frac{dx}{dt}\right)_{2,i} + 3.7\left(\frac{dx}{dt}\right)_{1,i} - 0.9\left(\frac{dx}{dt}\right)_{0,i}\right) \quad (65)$$

$$x_{yi} = x_{yi-1} + \frac{\Delta t}{2.4}\left(5.5\left(\frac{dy}{dt}\right)_{3,i} - 5.9\left(\frac{dy}{dt}\right)_{2,i} + 3.7\left(\frac{dy}{dt}\right)_{1,i} - 0.9\left(\frac{dy}{dt}\right)_{0,i}\right) \quad (66)$$

$$v_{xi} = v_{xi-1} + \frac{\Delta t}{2.4}\left(5.5\left(\frac{dv_x}{dt}\right)_{3,i} - 5.9\left(\frac{dv_x}{dt}\right)_{2,i} + 3.7\left(\frac{dv_x}{dt}\right)_{1,i} - 0.9\left(\frac{dv_x}{dt}\right)_{0,i}\right) \quad (67)$$

$$v_{yi} = v_{yi-1} + \frac{\Delta t}{2.4}\left(5.5\left(\frac{dv_y}{dt}\right)_{3,i} - 5.9\left(\frac{dv_y}{dt}\right)_{2,i} + 3.7\left(\frac{dv_y}{dt}\right)_{1,i} - 0.9\left(\frac{dv_y}{dt}\right)_{0,i}\right) \quad (68)$$

Preserve the past three values of v_{xi} and v_{yi}.

Loop from k = 0 to 2

$(dx/dt)_{k,i} = (dx/dt)_{k+1,i}$ (69)

$(dy/dt)_{k,i} = (dy/dt)_{k+1,i}$ (70)

$(dv_x/dt)_{k,i} = (dv_x/dt)_{k+1,i}$ (71)

$(dv_y/dt)_{k,i} = (dv_y/dt)_{k+1,i}$ (72)

End the loop

Set the new values

$(dx/dt)_{3,i} = v_{xi}$ (73)

$(dy/dt)_{3,i} = v_{yi}$ (74)

$(dv_x/dt)_{3,i} = (F_{xi}/m_i)$ (75)

$(dv_y/dt)_{3,i} = (F_{yi}/m_i)$ (76)

End the Loop from i = 1 to N

Output the new position and radius for the new state of the colloidal system.

Count the number of particles that have accumulated onto the collector.

End the Loop for the number of states.

Display the calculated states and display vital data to the display.

The above algorithm provides the necessary instructions to visualize the colloidal system on a computer. The following chapter describes the implementation of the algorithm on a personal computer.

T

comput

Microsc

[N] Rec

and appr

T

program

means all

mean out

n step ter

states of t

scientific

Th

DISP. The

communic

Th

simulator

mation:

1.

2.

IMPLEMENTATION

The algorithm outlined in chapter three was implemented on an PC - clone type computer utilizing the Intel X86 series microprocessor chip. The program was written in Microsoft Quick Basic version 4.5 [17] and was linked with an additional graphics library [18]. Requirements for the program are only a color video graphics adapter (VGA) monitor and approximately 200 kilobytes of memory.

The computer program is actually composed of three sub-programs. The first sub-program is the human-machine interface consisting of several pop up menus. The pop up menus allow a user flexibility to change the initial values defined in step one of the algorithm outlined in chapter three. The second sub-program consists of the algorithm defined in step ten in chapter three. This program does all of the calculations to generate the new states of the colloidal system. The third sub-program provides the graphic display allowing scientific visualization to take place.

The names of the three sub-programs are respectively, CSS, CSSRUN and CSSDISP. The acronym CSS stands for colloidal suspension simulator. Figure 4.1 shows the communication links between the three sub-programs.

The following discussion will provide instruction for using the colloidal suspension simulator software. The floppy disk included with this thesis contains the following information:

1. BIN: This directory contains the executable files that run the software. The names of the files residing in this directory are:
   A. CSS.EXE
   B. CSSDISP.EXE
   C. CSSRUN.EXE

2. INCLUDE: This directory contains the include files that hold common

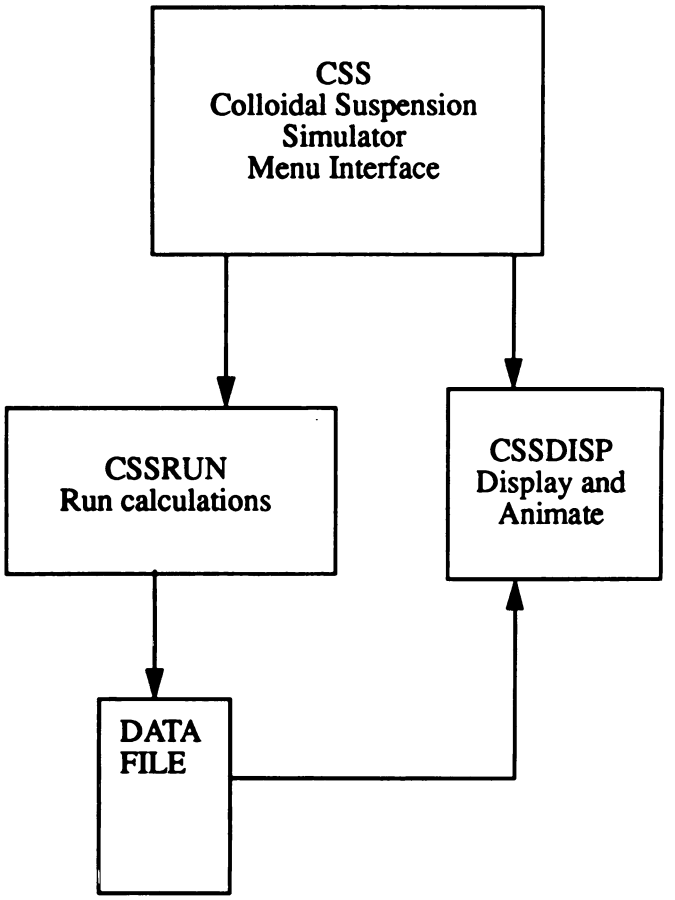


Figure 4.1 Communication links between the three sub - programs.

This figure shows the direction of flow between the CSS, CSSRUN and CSSDISP sub - programs.

information and initialization parameters that are needed to run the software. The names of the files residing in this directory are:

A. CSSCOM.INC

B. CSSINIT.INC

C. MENUPAR.INC

3. SOURCE: This directory contains all of the binary computer source code for use with a Quick Basic compiler. The names of the files residing in this directory are:

A. CSS.BAS

B. CSSDISP.BAS

C. CSSRUN.BAS

4. TXT: This directory contains all of the human-readable, ASCII computer source code listings of the software. The names of the files residing in this directory are:

A. CSS.TXT

B. CSSDISP.TXT

C. CSSRUN.TXT

The files necessary to run the software reside in the BIN directory. To install the software on a computer, the three executable files need to be copied from the BIN directory to a directory on the computer's hard drive.

To run the software, enter the command CSS. A menu will appear entitled, "Colloidal Suspension Simulator" with a second menu entitled, "Main Menu". To select any of the options listed in the main menu, the high-lighted letter or number need only be pressed. By

pressing the letter "H", for example, the help menu for the main menu will appear.

The main menu allows for the choice of a simulation type, to save or load a configuration and to display a previously calculated simulation. The first simulation involves the interactions between two types of particles, i. e., between particles of type A and particles of type B. The second simulation type involves the interaction between one type of particle and a fiber.

A configuration is the set of current values that describe information about the particles, the suspension medium, the fiber, the electrolyte, the surface parameters and the choice of a dynamic model. When the program is run for the first time, a default configuration is loaded. The default configuration is shown in table 4.1.

Once a simulation type is specified by pressing "1" or "2" at the main menu, a menu entitled either, "Particle A - Particle B" or "Particle A - Fiber" will be displayed. This menu will be referred to as the simulation menu here after. In either case, the selections for the simulation menu allow for the modification of material parameters, the modification of surface parameters, the choice of a dynamic model and a gateway for running the CSSRUN sub-program.

By pressing "2" at the simulation menu, the "Surface Parameter" menu will appear. This menu allows the pH of the medium to be changed and allows for the input of the zeta potential data for each particle or fiber.

By pressing "3" at the simulation menu, the "Dynamic Model" menu will appear. The three possible model types to run are named, DLVO Theory Model, The Acid/Base Model and the Random Model. Only the DLVO model is implemented at this time. The DLVO Theory Model uses the theory stating that the stability of a colloidal suspension is based on the sum of the electrostatic repulsions due to the overlap of electrical double layers plus the attractive potential due to the London - van der Waals forces.

By pressing "4" at the "Dynamic Model" menu, the time increment can be modified. The time increment specifies the time interval between the calculations of each state

Table 4.1 Default parameters used to initiate the CSS software.

Particle Information	Particle A	Particle B
Common Name	Iron Aluminum	Alumina
Chemical Name	FeAl	Al_2O_3
Shape	Spherical	Spherical
Density	5.56 gm/cm^3	3.97 gm/cm^3
Average Initial Velocity	30.0 μm/sec	30.0 μm/sec
Mean Diameter	1.0 μm	0.9 μm
Diameter Standard Deviation	0.001	0.001
Number of Particles	5	5
Hamaker Constant	3.08E-19 Joules	1.54E-19 Joules

Fiber Information	Fiber
Common Name	Alumina
Chemical Name	Al_2O_3
Shape	Cylindrical
Density	3.97 gm/cm^3
Diameter	25.0 μm
Hamaker Constant	1.54E-19 Joules

Medium Information	Medium
Common Name	Deionized Water
Chemical Name	H_2O
Density	1.0 gm/cm^3
Viscosity	0.01 gm/(cm sec)
Hamaker Constant	4.35E-20 Joules
Permeability	78.54
Temperature	25.0 Degrees Celsius

Electrolyte Information	Electrolyte
Common Name	Potassium Nitrate
Chemical Name	KNO_3
Concentration	0.0001 Normality

Surface Parameter Information	Surface Parameter
pH Level	8.0
Zeta Potential, particle A	5.0
Zeta Potential, particle B	18.00
Zeta Potential, Fiber	18.00
Time Increment	0.0000001 sec
Dynamic Model	DLVO

of the system.

Returning to the simulation menu and pressing "1" results in the "Material Parameters" menu to be displayed. The "Material Parameters" menu allows for the modification of information for particle A, for particle B, for the fiber, for the suspension medium and for the electrolyte.

By pressing "1" at the "Material Parameters" menu, the "Particle Information" menu appears. The following particle information may be modified:

1. Common Name
2. Chemical Name
3. Shape
4. Maximum Diameter
5. Mean Diameter
6. Diameter Standard Deviation
7. Density
8. Number of Particles
9. Hamaker Constant

If the "Particle A - Fiber" simulation was chosen, then pressing "2" at the "Material Parameters" menu causes the "Fiber Information" menu to appear. The following fiber data may be modified:

1. Common Name
2. Chemical Name
3. Shape
4. Diameter
5. Density
6. Hamaker Constant

By pressing "3" at the "Material Parameters" menu, the "Medium Information" menu appears. The following medium data may be modified:

1. Common Name
2. Chemical Name
3. Density
4. Viscosity
5. Hamaker Constant
6. Permeability
6. Temperature

By pressing "4" at the "Material Parameters" menu, the "Electrolyte Information" menu appears. The following electrolyte data may be modified:

1. Common Name
2. Chemical Name
3. Concentration

Returning to the simulation menu and pressing "R", the "Run Model" menu will appear. An information box will appear at the bottom of the screen and will state: "Enter the Output File Name (with no extension): ". The requested file name will be used to write out the x and y positions of the colloidal particles at each time frame. The user may press enter without typing a file name and will be prompted with "Do You Wish to Quit [N]?". Upon entering a valid output file name at the initial prompt, a second prompt will appear stating: "Enter the Number of Frames to be Calculated:". The number of frames to be calculated correspond to the number of states of the system that will be animated with the CSSDISP sub program.

Once a valid number of frames to be calculated has been entered, the CSSRUN sub program will perform the calculations to generate each frame. Status messages will be displayed to the screen to allow the user to keep track of how the calculations are progressing. Upon successful completion of the sub program, the user will be asked to press "C" to continue. The file containing the newly calculated data MUST be written down and remembered at this point. The name of the data file will be used later in the display sub program.

The main menu will reappear. By pressing "D" at the main menu, the "Display Model" menu will appear. A prompt at the bottom of the screen will appear stating, "Enter the Input File Name (with no extension):". The user may quit from this menu by pressing enter without entering a file name. The input file name that the software is looking for is the name of a file that was created by the CSSRUN sub program. If a valid file name is entered, the "Colloidal Suspension Simulator" display menu will appear. The right column of the screen displays the choice of dynamic model, the pH level of the medium, the zeta potentials of each material, the Debye length, the concentration, the time increment between each frame and the current frame number.

By pressing any key, the simulation will begin. The simulation can be paused by

pressing the "enter" key or stopped entirely by pressing any other key. The up and down "arrow" keys adjust the replay speed of the simulation, level 20 being the fastest speed and level 0 being the slowest speed. The simulation may be replayed again and again by pressing the space bar.

Simulations that have been run previously may be viewed again by choosing the desired file at the "Display Model" menu.

RESULTS / DISCUSSION

The algorithm derived in chapter three requires some empirical data, namely, the zeta potential data. The colloidal suspension software requires the user to input the pH of the medium and the corresponding zeta potentials of each particle or fiber. Zeta potential data was obtained for the intermetallic, Fe-40Al, particles and for two types of Alumina, Al_2O_3 - FP and Al_2O_3 PRD-166. The data was obtained at E. I. DuPont in Wilmington, Delaware under the supervision of Dr. Rulon Johnson and Mr. Jerry Hughes. The zeta potentials were measured using the acoustophoresis technique on an electrokinetic sonic amplitude (ESA) device engineered by Matec Inc. The data obtained from the Matec 8000 is displayed in Figure 5.1, in Figure 5.2 and in Figure 5.3. The zeta data for the Al_2O_3 fiber obtained at Dupont, however, does not agree with the known zeta potential data for Alumina. The inconsistency with the Al_2O_3 fiber was most likely attributed to the large, discontinuous fiber sizes that were not entirely eliminated by the grinding process that was used. The zeta potential data shown in Figure 5.4 and in Figure 5.5 were performed with greater accuracy using a Matec 8000 by Brett Wilson [19].

The primary equations that determine the stability of the colloidal suspension according to the DLVO theory were found in the literature and were presented in chapter two by equations (19) and (28). The corresponding equations for the repulsive force, equations (35) through (37), and the attractive force, equations (38) through (41), were derived from the equations given in the literature.

The algorithm described in chapter three was based on the general method for solving a many body system. The algorithm was tailored and customized for the explicit intent of solving the many body problem for a colloidal suspension system.

The implementation of the algorithm in the form of the colloidal suspension simulator (CSS) software was the intended result of this thesis. The software in its delivered form provides a flexible tool for understanding colloidal suspensions. The CSS software

Fe-40A1

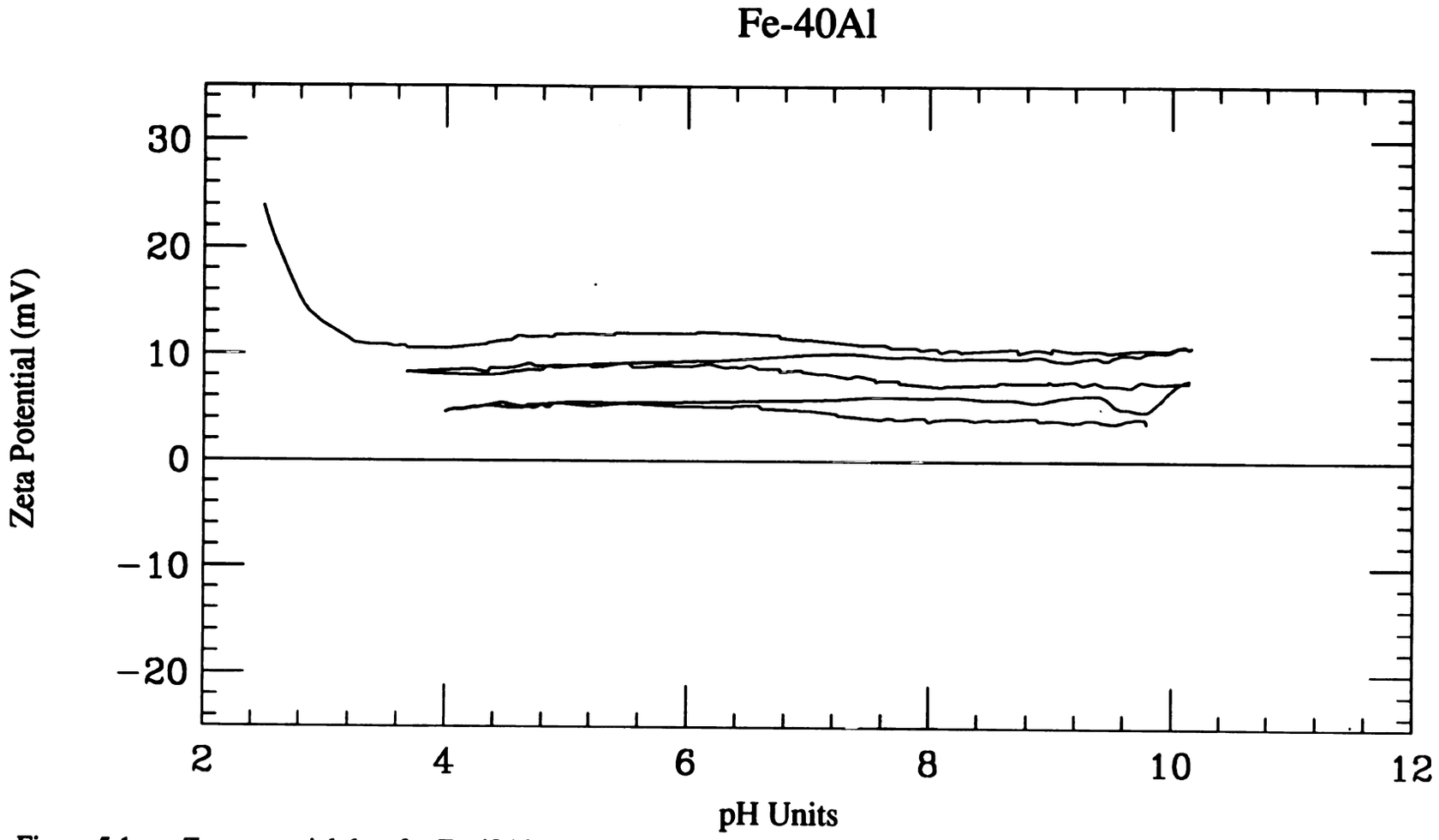


Figure 5.1 Zeta potential data for Fe-40Al.

The measured data was obtained using 5.0 wt% Fe-40Al (0.935 volume%) powder dispersed into 0.001N KNO_3.

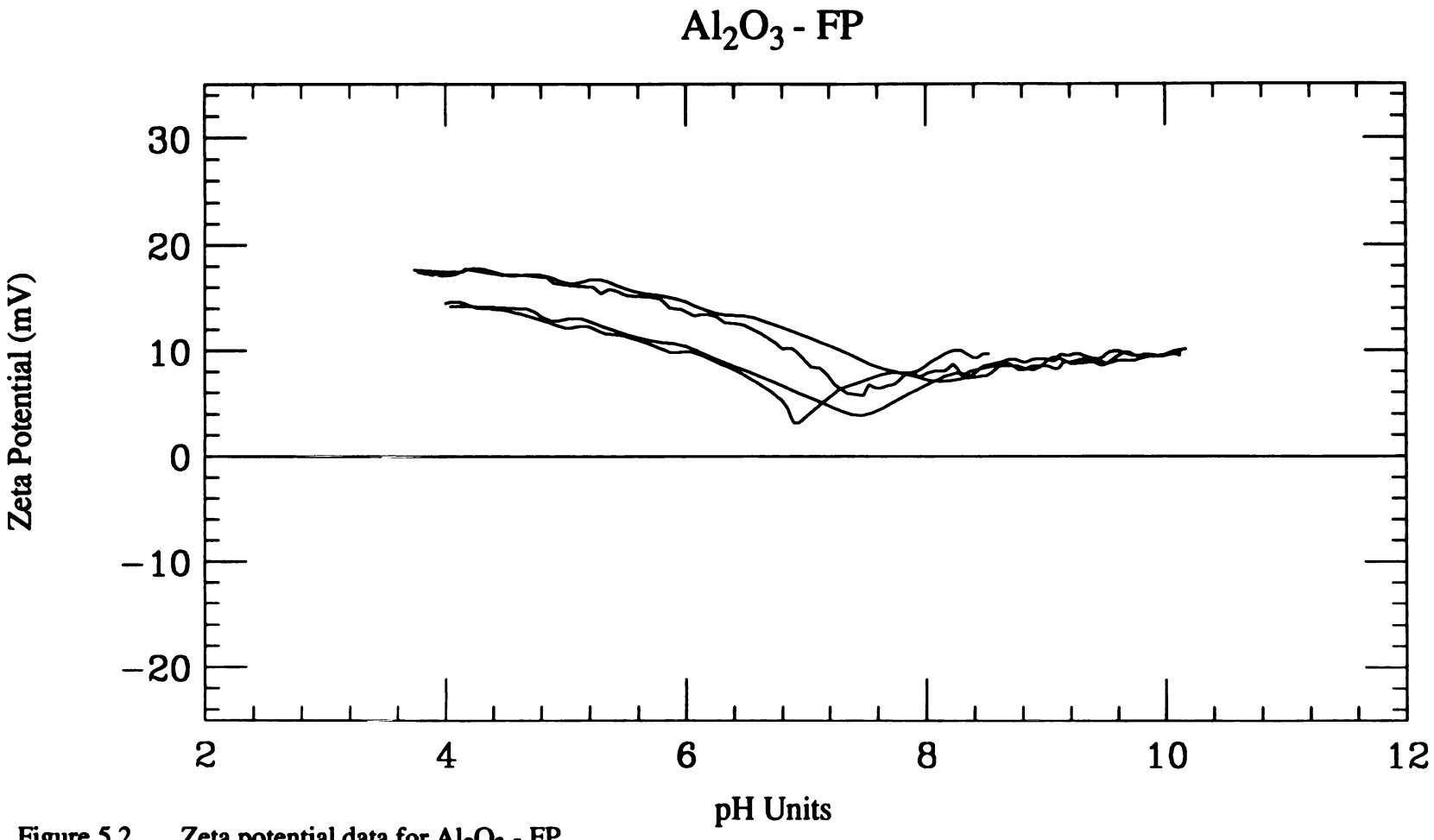


Figure 5.2 Zeta potential data for Al_2O_3 - FP.

The measured data was obtained using 1.075 wt% (0.272 volume%) Al_2O_3 -FP dispersed into 0.001 N KNO_3.

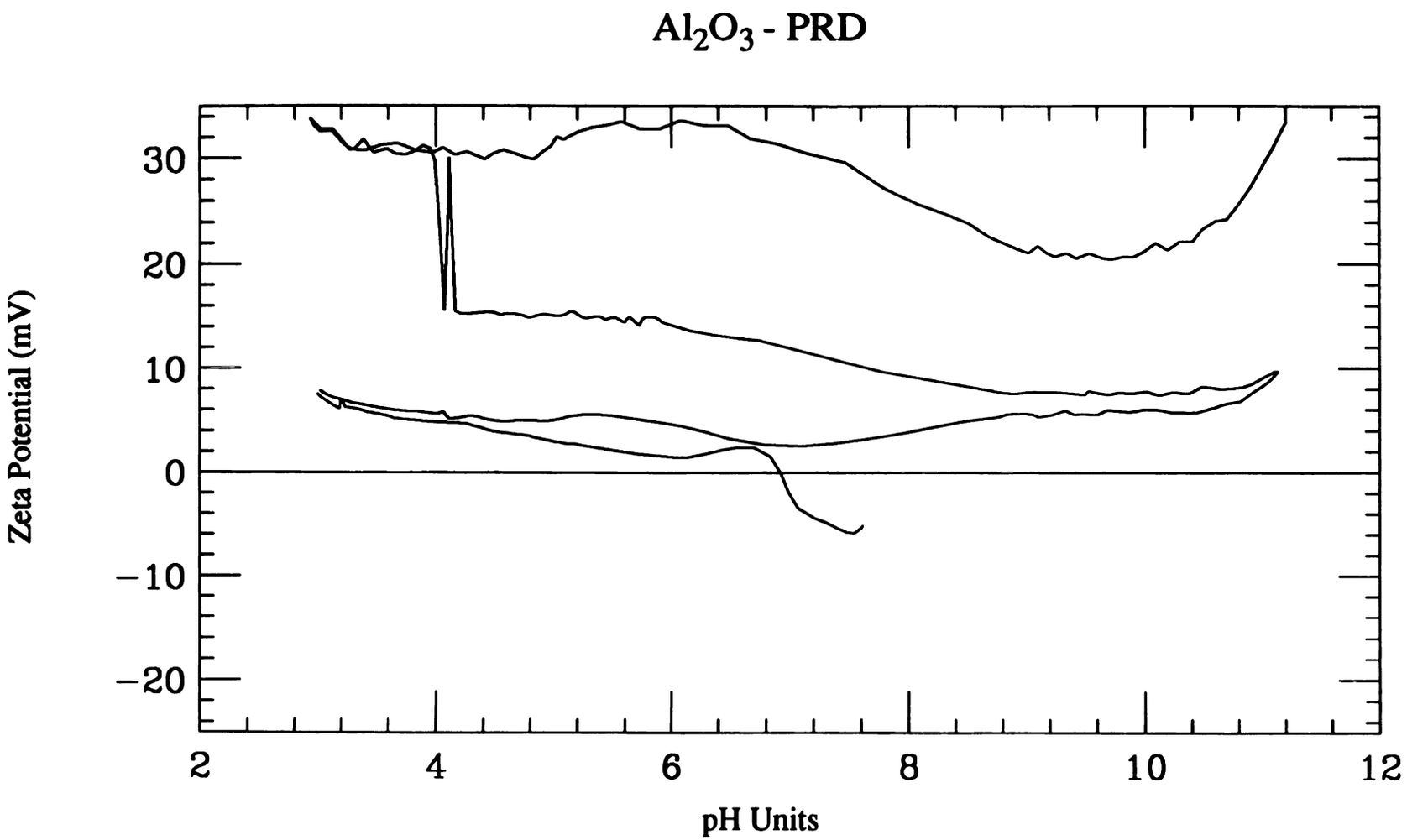


Figure 5.3 Zeta potential data for Al_2O_3 - PRD-166.

The measured data was obtained using 1.075 wt% (0.272 volume%) Al_2O_3 - PRD-166 in 0.001N KNO_3.

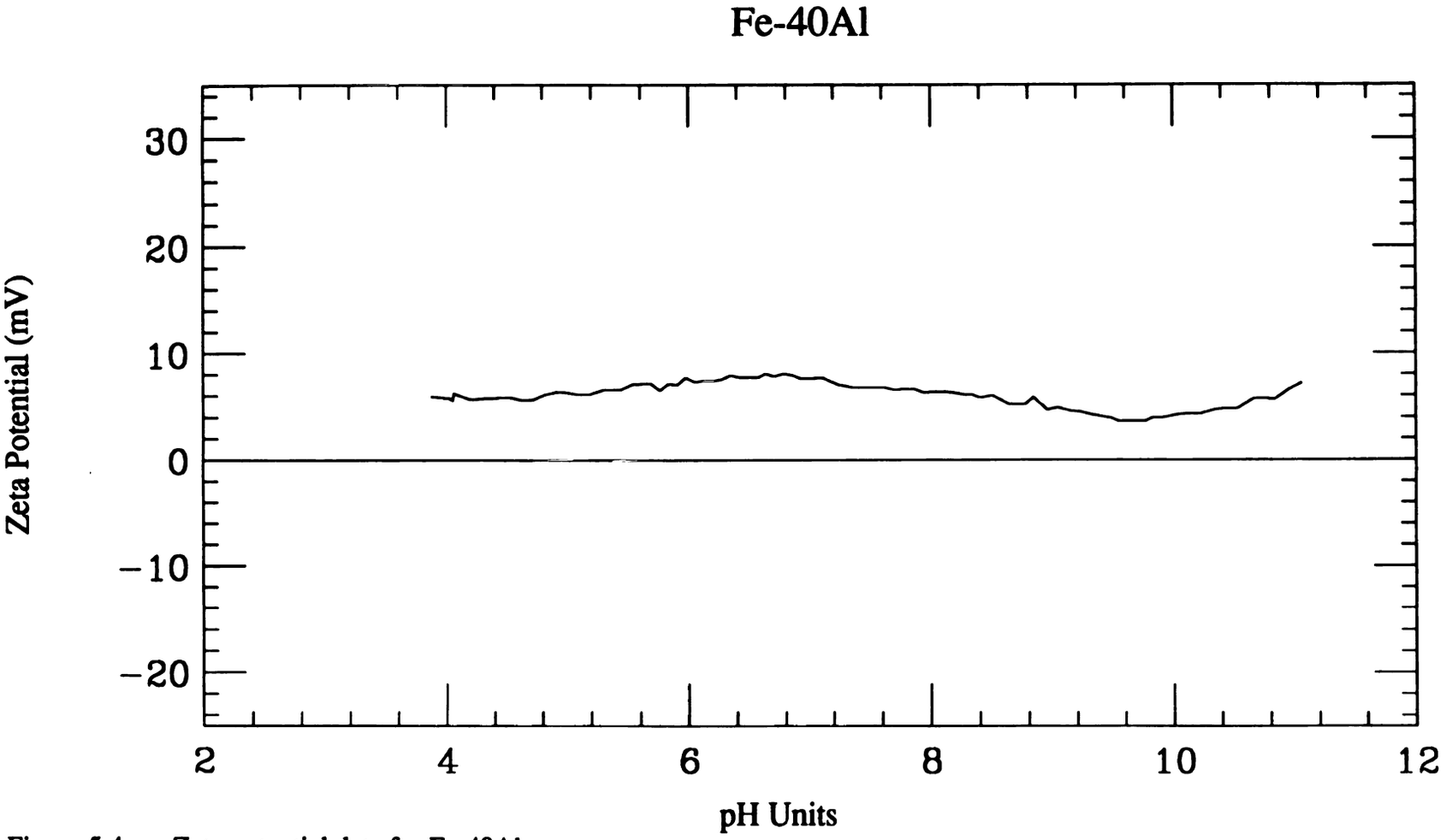


Figure 5.4 Zeta potential data for Fe-40Al.

The measured data was obtained using 0.2 volume% Fe-40Al powder dispersed into 0.001N KNO_3. This data was measured by Bret Wilson [19].

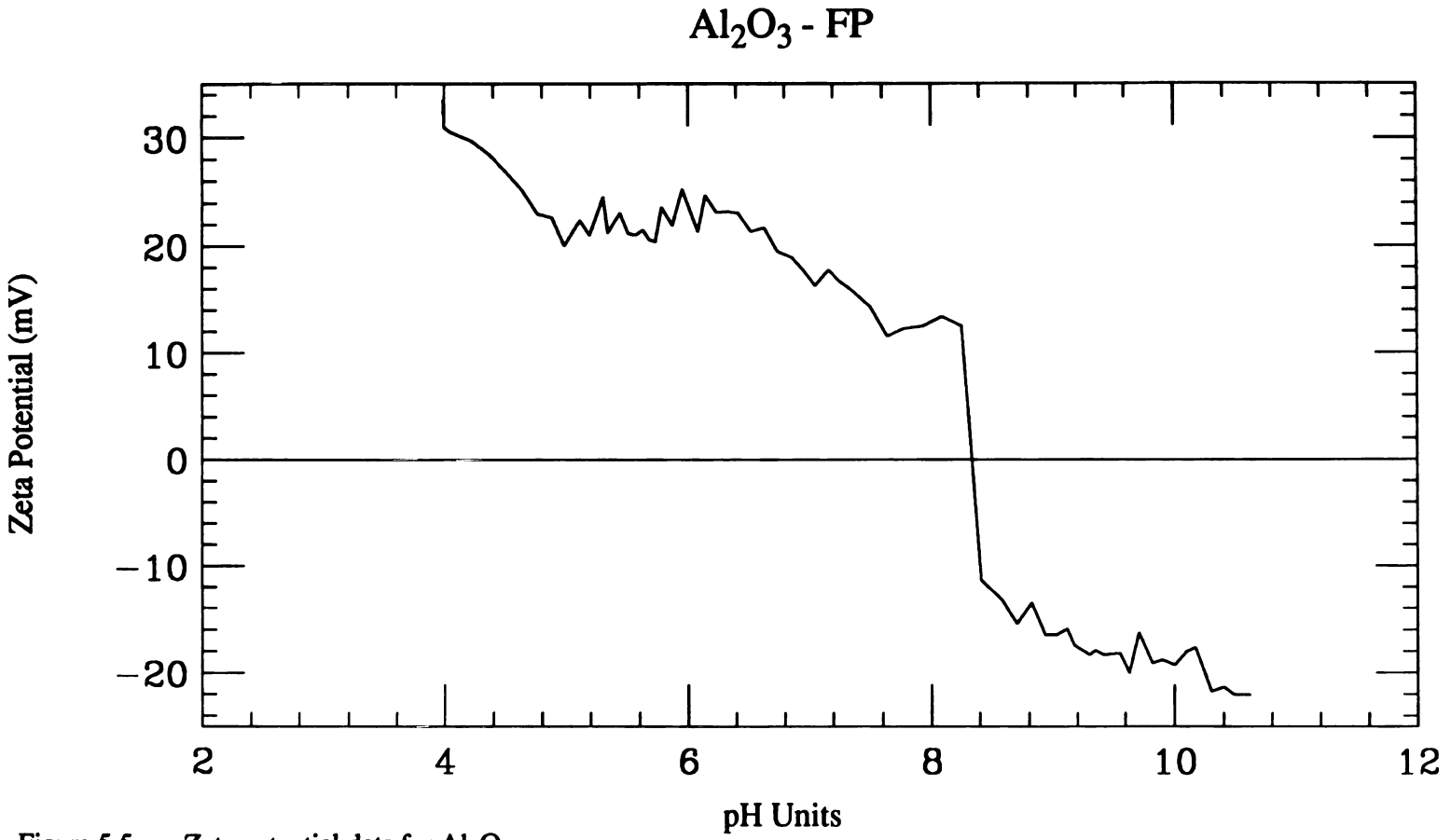


Figure 5.5 Zeta potential data for Al_2O_3.

The measured data was obtained using 0.5 volume% Al_2O_3 dispersed into 0.001 N KNO_3. This data was measured by Bret Wilson [19].

presents all of the necessary parameters and a priori data that are required to define the system. Any of the parameters that describe the system may be modified easily and efficiently to maximize the ability to control the system in any desired way. Upon finding acceptable parameters that suit a users needs, the entire configuration of the system as well as the actual simulation data may be recorded. The flexibility and control of the colloidal system is a major advantage for using the CSS software versus conducting tedious and perhaps costly experimental procedures for understanding the suspension mechanics.

Five tests were run with the CSS software to determine the influence of certain parameters on the outcome of a simulation. Each of the tests resulted in several data files that are included in this thesis on two floppy disks labeled, "Test Data". The test data may be viewed by copying the contents of each "test" directory on the floppy disk to the location on the computer's hard drive where the CSS software resides. An example of the format of a data file that was generated by the CSSRUN sub - program is displayed in Figure 5.6. The data file shown in Figure 5.6 contains the state information for three particles for four time frames. The following discussion will outline the conducted tests.

The first test was an attempt to determine how the time increment between calculations affects the dynamics of the simulation. The default parameters defined in table 4.1 were used. The names of the output files are, respectively, A, B and C and each simulation ran for 500 frames. The data in A used a time increment of 0.001 seconds. The data in B used a time increment of 0.01 seconds. The data in C used a time increment of 0.1 seconds. Upon viewing each simulation with the CSSDISP software, it was evident that the time increment is quite crucial for accurate position calculations to be determined. The error for determining the new position at each time frame starts out very small, since the initial positions are given, but the error appears to propagate as time progresses. The smaller time increment minimizes the accumulation of errors during the calculation of each time frame.

The second test was an attempt to determine the affect of a variable concentration level while maintaining the other parameters. The default parameters defined in table 4.1

```
CSS DATA FILE
3
0.500706, 15
0.500760, 15
0.450549, 10
0
93.000000, 57.500000
78.000000, 45.500000
78.000000, 39.500000
1
92.966286, 57.473743
77.973236, 45.557690
77.989143, 39.527519
2
92.932709, 57.447628
77.946426, 45.615337
77.977898, 39.554565
3
92.899361, 57.421753
77.919540, 45.672920
77.966003, 39.580826
```

Figure 5.6 Data file created with the CSSRUN software.

This figure displays an example data file that is generated by the CSSRUN sub - program and is read by the CSSDISP sub - program. The first number is the number of particles, followed by the radius and color number of each of the three particles. The rest of the data format is repeated showing the frame number followed by the x and y positions of each particle at that time frame.

were used. The data files for this test are called, respectively, D, E, F and G.The time increment was held at 0.001 seconds and each simulation ran for 500 time frames. The data in D used a concentration of 0.001 N. The data in E used a concentration of 0.01 N. The data in F used a concentration of 0.1 N. The data in G used a concentration of 1.0 Normality. No noticeable differences were apparent between each of the simulations. This is an unexpected result, and provides the first indication that the dynamics of the software is not behaving as in nature. Colloidal theory predicts that as the electrolyte concentration increases, the double layer length, $1/\kappa$, will decrease. The decrease in the double layer length reduces the magnitude of the repulsive force between the colloids. In this case, the colloids are able to flocculate and form agglomerates. This phenomenon was not observed in simulations D, E, F or G.

Test number three was an experiment to determine how a variable particle size would affect the colloidal system. The default parameters defined in table 4.1 were used except that it was discovered experimentally that the a time increment of 0.00001 seconds was required. The data files for test three are called, respectively, H, I, J, K and L. The data in H used a mean diameter of 0.1 microns and ran for 20 time frames. The data in I used a mean diameter of 0.5 microns and ran for 50 frames. The data in J used a mean diameter of 0.3 microns and ran for 50 frames. The data in K used a mean diameter of 0.15 microns and ran for 200 frames. The data in L used a mean diameter of 0.50 microns and ran for 10 frames. In experiments H and L, the particles move very rapidly off the screen. In experiments I and J, the particles are relatively motionless. In experiment K, the particles are slightly mobile. These experiments suggest that as the particle size becomes smaller, a smaller time increment is required to control the dynamics of the suspension.

Test number four was an experiment to determine how a change in the viscosity of the medium would affect the colloidal system. The default parameters defined in table 4.1 were used except that the particles average initial velocities were given a value of 50.0 microns/second. The data file for this test is called M. The viscosity of the medium, the

deionized water, was intentionally set to 999 gm/(cm sec). The particles moved very slowly as the hydrodynamic forces would suggest.

Test number five was used to observe the affects of the suspension due to a variable pH level and consequently, variable zeta potential data. The default parameters defined in table 4.1 were used. The data files for this test are called, respectively, N, O and P. Each simulation ran for 300 time frames. The data in N used a pH level of 5.0 with zeta potentials of 5.0 mV and 18.0 mV for Fe-40Al and Al_2O_3, respectively. The data in O used a pH level of 7.8 with zeta potentials of 5.0 mV and 12.0 mV for Fe-40Al and Al_2O_3, respectively. The data in P used a pH level of 10.0 with zeta potentials of 5.0 mV and -19.0 mV for Fe-40Al and Al_2O_3, respectively. None of the data files exhibited expected results. This provides the second indication that the dynamics of the software is not behaving as nature. For the suspension under investigation, colloidal theory predicts that as the pH level varies causing the zeta potential data to fluctuate, the stability of the suspension should change.

The test data files show some of the experimental capabilities that are possible using the CSS software for probing many unanswered questions concerning colloidal suspensions. It would appear that many combinations of changing one, two or more initial parameters to conduct an experiment are possible with the CSS software. The actual number of possible experiments is calculated by summing the combinations of all of the variables taken one at a time, two at a time,..., n at a time, where n is the number of variables. The important changeable variables with their degrees of freedom are listed below in table 5.1. The total number of experiments possible using 19 variables is 524,287. This large number of combinations shows how the colloidal suspension simulator can be used as an important experimental tool.

Table 5.1 Number of changeable variables using the CSS software.

Variable	Degrees of Freedom
Mean Diameter	2
Initial Velocity	2
Number of Particles	2
Hamaker Constant	4
Density	3
Viscosity	1
Temperature	1
Concentration	1
pH Level	2
Time Increment	1
(Dynamic Model)	(3)
Total: 19 Variables	

CONCLUSIONS

The zeta potential data obtained at Dupont as described in chapter five provided the first clues of the surface characterization of the Iron Aluminum particles and the Al_2O_3 fiber. The zeta potential data shown in Figures 5.4 and 5.5 are the most accurate zeta potential data obtained at the present time.

The repulsive and attractive force equations, equations (35) through (37) and equations (38) through (41), provide an alternative method for viewing a colloidal system.

The fundamental forces arising in a colloidal suspension were defined. A detailed description for the solution to a many body dynamical system was tailored to describe the dynamics of a colloidal suspension system. The algorithm for describing the colloidal suspension was implemented on a personal computer.

The colloidal suspension simulator (CSS) software was tested by modifying certain initial parameters and observing the animated suspended particles. The choice of a small time increment was found to be an important factor for ensuring accuracy during calculations. Experimentation with the CSS showed that as particle diameters become smaller than 0.1 microns, a time increment on the order of 10^{-4} or smaller is required. The CSS software is a working prototype, but with some discrepancies. No noticeable differences were observed between the experimental data for different electrolyte concentrations. Changes in the surface parameters produced no noticeable change to the system for different experiments. These two points suggest that the equations describing the electrodynamic forces need some attention. The movement of the colloidal particles were severely damped when the viscosity of the medium was increased which agrees with the hydrodynamic expectations.

The CSS software is flexible and provides thousands of possible ways to analyze and to understand the dynamics of colloidal suspensions.

RECOMMENDATIONS

The computer program does not function as well as desired. Problems encountered with the implementation and recommendations include the following:

- Particle collisions were not included in the computer program. By not properly accounting for collisions, the particles are able to penetrate each other and the integrity of the net forces is corrupted.
- Units consistency needs to be further verified to ensure that the order of magnitude difference between forces, velocities and positions are within reason.
- The time increment between calculations could be chosen with greater sophistication. A variable time step might eliminate the problem that arises with particle impacts of three or more. Also, a consistently small enough time step will eliminate a particle "jumping" over another particle that possibly prevents some instances of collisions from occurring at all.
- The coagulation of particles needs to be thought out and handled appropriately. One method could be such that if two particles of radius r_1 and r_2 coagulate, then the two particles coalesce into a new larger particle of radius $r_1 + r_2$. Another possible method is to use rigid body mechanics for an agglomerate.

Once proper implementation of the colloidal suspension simulator has been achieved, useful experiments may be conducted. For specified parameters, the stability ratio of the suspension may be used to determine the critical coagulation concentration. The stability ratio is the total number of collisions divided by the total number of collisions resulting in adhesion. The pH level regions at which the particles flocculate, becoming unstable may be determined. An accumulative mass function could be added to the program to keep track of the mass per unit area per unit time that collects onto the fiber. An interesting three dimensional graph would be to plot time versus pH versus the amount of accumulated mass on the collector. The accumulated mass functionality would provide an

indication of the rate at which the particles collect onto the fiber for a specified pH level. Figure 7.1 depicts such a possible image.

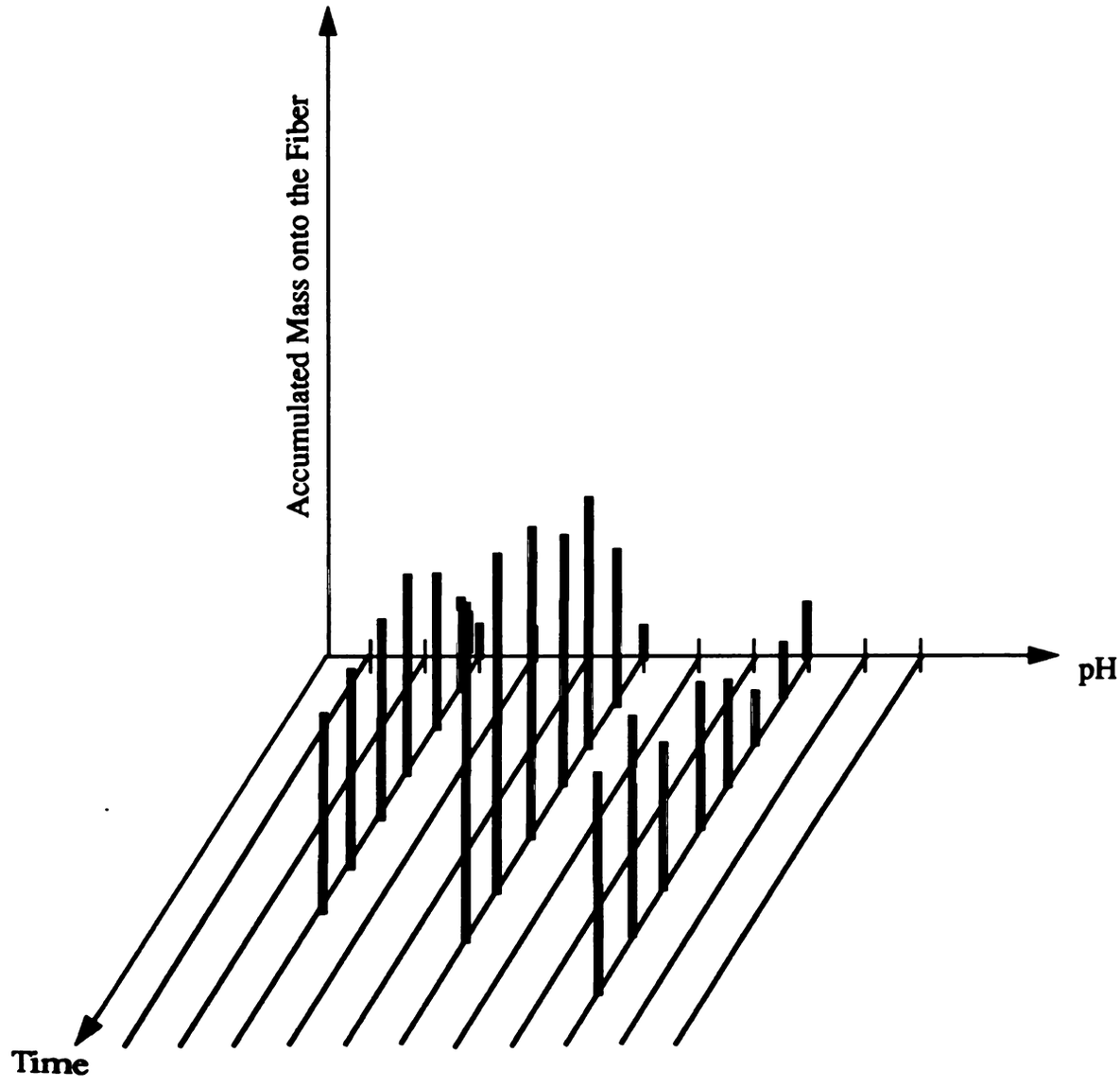


Figure 7.1 Depiction of accumulated mass.

This figure shows an example of data that could be calculated using the colloidal suspension simulator.

LIST OF REFERENCES

1. Crimp, M. A. and Crimp, M.J., REF Proposal, July 1990, "Fiber Electrophoretic Deposition Processing of FeAl/Al_2O_3 Intermetallic Matrix Composite".

2. Heimenz, Principles of Colloid and Surface Chemistry, Marcel Dekker, New York, 2nd edition,1986.

3. Ross, Sydney and Morrison, Ian Douglas, Colloidal Systems and Interfaces, John Wiley and Sons, New York, 1988.

4. Everett, D. H., Basic Principles of Colloid Science, Royal Society of Chemistry, London, 1988.

5. Wiese, G. R. and Healy, T. W., *Transactions of the Faraday Society*, 1970, 66, 490 - 499, "Effect of Particle Size on Colloid Stability".

6. Hirtzel, C. S. and Rajagopalan, Raj, Colloidal Phenomena: Advanced Topics, Noyes Publications, New Jersey, 1985.

7. Mahanty, J. and Ninham, B. W., Dispersion Forces, Academic Press, Inc., London, 1976.

8. Mysels, Karol J., Introduction to Colloid Chemistry, Robert E. Krieger Publishing Company, Huntington, New York, 1978.

9. Shaw, Duncan J., Introduction to Colloid and Surface Chemistry. Butterworths, London, 1966.

10. Derjaguin, B. V., *Discussions of the Faraday Society*,Vol. 18, 85 - 98, 1954, "A Theory of the Hetercoagulation, Interaction and Adhesion of Dissimilar Particles in Solutions of Electrolytes."

11. Hogg, R., Healy, T. w. and Fuerstenau, D. W., *Transactions of the Faraday Society*, 62, 1638, 1966, "Mutual Coagulation of Colloidal Dispersions".

12. Bell, G. M., Levine, S. and McCartney, L N., *Journal of Colloid and Interface Science*, Vol. 33, No. 3, 335 - 359, July 1970, "Approximate Methods of Determining the Double - Layer Free Energy of Interaction Between Two Charged Colloidal Spheres".

13. Bell, G.M. and Peterson, G. C., *Journal of Colloid and Interface Science*, Vol. 41, No. 3, 342 - 366, December 1972, "Calculation of the Electric Double - Layer Between Unlike Spheres".

14. Mahanty, J. and Ninham, B. W., Dispersion Forces, Academic Press, Inc., London, 1976.

15. Hamaker, H. C., *Physica* 4, 1058 - 1072, 1937, "The London - van der Waals Attraction Between Spherical Particles".

16. Beer, Ferdinand P. and Johnston, E. Russel, Vector Mechanics for Engineers: Dynamics, McGraw-Hill Book Company, Ney York, 1988.

17. Hammerly, Wayne M. and Hanlin III, Thomas G., ADVAS Proramming Library, Shareware, 1988.

18. Microsoft Corporation, QuickBASIC 4.50, 1985-1988.

19. Wilson, Brett, Masters Thesis, "Prediction of Colloidal Suspension Stability for SiC/Si_3N_4 and FeAl/Al_2O_3 Fiber Systems Using Material and System Parameters",Michigan State University, 1992.

GENERAL REFERENCES

1. Heimenz, Principles of Colloid and Surface Chemistry, Marcel Dekker, New York, 2nd edition,1986.

2. Crimp, M. A. and Crimp, M.J., REF Proposal, July 1990, "Fiber Electrophoretic Deposition Processing of FeAl/Al_2O_3 Intermetallic Matrix Composite".

3. Hamaker, H. C., *Physica* 4, 1058 - 1072, 1937, "The London - van der Waals Attraction Between Spherical Particles".

4. Hogg, R., Healy, T. w. and Fuerstenau, D. W., *Transactions of the Faraday Society*, 62, 1638, 1966, "Mutual Coagulation of Colloidal Dispersions".

5. Bierman, Arthur, *Journal of Colloid and Interface Science*, 10, 231 - 245, 1955, "Electrostatic Forces Between Nonidentical Colloidal Particles".

6. Barouch, Eytan, Matijevic, Egon, Ring, Terry A. and Finlan, J. Michael, *Journal of Colloid and Interface Science*, Vol. II, No. 1, October 1978, "Hetercoagulation II. Interaction Energy of Two Unequal Spheres".

7. Bell, G. M., Levine, S. and McCartney, L N., *Journal of Colloid and Interface Science*, Vol. 33, No. 3, 335 - 359, July 1970, "Approximate Methods of Determining the Double - Layer Free Energy of Interaction Between Two Charged Colloidal Spheres".

8. Bell, G.M. and Peterson, G. C., *Journal of Colloid and Interface Science*, Vol. 41, No. 3, 342 - 366, December 1972, "Calculation of the Electric Double - Layer Between Unlike Spheres".

9. Schenkel, J. K. and Kitchener, *Transactions of the Faraday Society*, September 1959, 161 - 173, "A Test of the Derjaguin - Landau - Verwey - Overbeek theory with a Colloidal Suspension".

10. Hall, Simon Berners, Duffield, John Ralph and Williams, David Raymond, *Journal of Colloid and Interface Science*, Vol. 143, No. 2, 416 - 422, May 1991, "A Stochastic Computer Simulation of Emulsion Coalescence".

11. Derjaguin, B. V., *Discussions of the Faraday Society*,Vol. 18, 85 - 98, 1954, "A Theory of the Hetercoagulation, Interaction and Adhesion of Dissimilar Particles in Solutions of Electrolytes."

12. Wiese, G. R. and Healy, T. W., *Transactions of the Faraday Society*, 1970, 66, 490 - 499, ""Effect of Particle Size on Colloid Stability".

13. Matijevic, Egon, *Pure and Appl. Chem.*, Vol. 53, 2167 - 2179, "Interactions in Mixed Colloidal Systems (Hetercoagulation, Adhesion, Microflotation)".

14. Tamura, Hiroki, Matijevic, Egon and Meites, Louis, *Journal of Colloid and Interface Science*, Vol. 92, No. 2, 303 - 314, April 1983, "Adsorption of Co^{2+} Ions on Spherical Magnetite Particles".

15. Rajagopalan and Chu, Richard Q., *Journal of Colloid and Interface Science,* Vol. 86, No. 2, 299 - 317, April 1982, "Dynamics of Adsorption of Colloidal Particles in Packed Beds".

16. Derjaguin, V. M., Muller and Toporov, Yu. P., *Journal of Colloid and Interface Science*, Vol. 53, No. 2, 314 - 326, November 1975, "Effect of Contact Deformations on the Adhesion of Particles".

17. Kar, G, Chander, S. and Mika, T. S., *Journal of Colloid and Interface Science*, Vol. 44, No. 2, August 1973, 347 - 355, "The Potential Energy of Interaction Between Dissimilar Electrical Double Layers".

18. Visser, J., *Advan. Colloid Interface Sci.*, 3, 331 - 363, 1972, "On Hamaker Constants: A comparison Between Hamaker Constants and Lifshitz - van der Waals Constants".

19. Verwey, E. J. W., Overbeek, J. T. G., Theory of the Stability of Lyophobic Colloids, Elsevier, Amsterdam, 1948.

20. Derjaguin, B. V., Landau, L. D., *Acta Physichim*, URSS, 14, 633, 1941.

21. Ross, Sydney and Morrison, Ian Douglas, Colloidal Systems and Interfaces, John Wiley and Sons, New York, 1988.

22. Russel, William B., The Dynamics of Colloidal Systems, The University of Wisconsin Press, 1987.

23. Vold, Robert and Vold, Marjorie J., Colloid and Interface Chemistry, Addison - Wesley Publishing Company, Inc., Reading, Massachusetts, 1983.

24. Mysels, Karol J., Introduction to Colloid Chemistry, Robert E. Krieger Publishing Company, Huntington, New York, 1978.

25. Shaw, Duncan J., Introduction to Colloid and Surface Chemistry, Butterworths, London, 1966.

26. Edited by Kerker, Milton, Zettlemoyer, Albert C. and Rowell, Robert L., Colloid and Interface Science Vol. I: Plenary and Invited Lectures, Academic Press Inc., London, 1988.

27. Everett, D. H., Basic Principles of Colloid Science, Royal Society of Chemistry, London, 1988.

28. Lynch, C. T. and Kershaw, J. P., Metal Matrix Composites, CRC Press, The Chemical Rubber Co., Cleveland, Ohio, 1988.

29. Mahanty, J. and Ninham, B. W., Dispersion Forces, Academic Press, Inc., London, 1976.

30. Marciniak, Andrzej, Numerical Solutions of the N - Body Problem, D. Reidel Publishing Company, Dordrecht, Holland, 1985.

31. Becher, Paul, Dictionary of Colloid and Surface Science, Marcel Dekker, Inc., New York, 1990.

32. Hirtzel, C. S. and Rajagopalan, Raj, Colloidal Phenomena: Advanced Topics, Noyes Publications, New Jersey, 1985.

33. Crimp, Melissa J., Masters Thesis, Case Western Reserve University, 1985, "Colloidal Characterization of Silicon Carbide and Silican Nitride".

34. Handbook of Chemistry and Physics, 69th Edition 1988 - 1989, CRC Press, Inc., Florida, 1989.

35. Mattuck, Richard, D., A Guide to Feynman Diagrams in the Many - Body Problem, McGraw - Hill Book Company, London, 1967.

APPENDIX A

CSS PROGRAM LISTINGS

```
DEFINT A-Z

'-----------------------------------------------

'===============================================
' Program Name: CSS
'
' File Name: CSS.BAS
'
' Description: This is the main routine for the colloidal
' suspension simulator. This routine collects
' the information to specify the initial conditions
' to run a simulation.
'
' Author: Peter T. Robinson
'
' Date: January 24, 1992
'
' Revision History:
' None
'
'===============================================
'
'===============================================
' Declare the subroutines
'===============================================
'
DECLARE SUB CLEARINFOBOX ()
DECLARE SUB DISPHELP (HELPMES$())
DECLARE SUB DISPDYNMODEL ()
DECLARE SUB DISPELECTRO1 ()
DECLARE SUB DISPERR (MESSAGE$())
DECLARE SUB DISPFIBER ()
DECLARE SUB DISPINFOBOX ()
DECLARE SUB DISPPARTA ()
DECLARE SUB DISPPARTB ()
DECLARE SUB DISPMAIN ()
DECLARE SUB DISPMAINH ()
DECLARE SUB DISPMATERIAL1 ()
DECLARE SUB DISPMATERIAL1H ()
DECLARE SUB DISPMEDIUM1 ()
DECLARE SUB DISPSIM1 ()
DECLARE SUB DISPSIM1H ()
DECLARE SUB DISPSURFACE1 ()
DECLARE SUB RUNDISPLAY ()
DECLARE SUB DYNMODEL ()
DECLARE SUB ELECTRO1 ()
DECLARE SUB FIBER ()
DECLARE SUB GETDYNMODEL (OPT%)
DECLARE SUB GETELECTRO1 (OPT%)
DECLARE SUB GETFIBER (OPT%)
DECLARE SUB GETFILENAM (FILENAM$)
DECLARE SUB GETMAIN (MODEL)
DECLARE SUB GETMATERIAL1 (OPT%)
DECLARE SUB GETMEDIUM1 (OPT%)
DECLARE SUB GETPARTA (OPT%)
DECLARE SUB GETPARTB (OPT%)
DECLARE SUB GETSIM1 (OPT%)
DECLARE SUB GETSURFACE1 (OPT%)
DECLARE SUB HELPDYN ()
DECLARE SUB HELPMAIN ()
DECLARE SUB HELPELECTRO ()
DECLARE SUB HELPFIBER ()
DECLARE SUB HELPMATRL ()
DECLARE SUB HELPMEDIUM ()
DECLARE SUB HELPPARTA ()
DECLARE SUB HELPPARTB ()
DECLARE SUB HELPSIM ()
DECLARE SUB HELPSURFACE ()
DECLARE SUB LOADATA ()
DECLARE SUB MATERIAL1 ()
DECLARE SUB MEDIUM1 ()
DECLARE SUB PARTA ()
DECLARE SUB PARTB ()
DECLARE SUB READSTAT ()
DECLARE SUB SIM1 ()
DECLARE SUB SIM1MENU ()
```

```
DECLARE SUB SAVEDATA ()
DECLARE SUB SURFACE1 ()
DECLARE SUB TRANSDATA ()
DECLARE SUB TRANSDATAG ()
DECLARE SUB WRITESTAT ()
'=============================================================

'$INCLUDE: 'CSSCOM.INC'

ON ERROR GOTO ErrorHandler

DIM SHARED HELPMES$(1 TO 19)
DIM SHARED MESSAGE$(1 TO 19)

'=============================================================
' Check to see if this is the first time the program has
' been run
'=============================================================

CALL READSTAT

IF INITDAT = 1 THEN
 CALL TRANSDATAG
ELSE
 '$INCLUDE: 'CSSINIT.INC'
END IF

DO

 DONE = 0

'=============================================================
' Display the Main menu
'=============================================================

 CALL DISPMAIN

'=============================================================
' Accept and verify the input to the main menu.
'=============================================================

 CALL GETMAIN(SIMNUM)

'=============================================================
' Run the selected Simulation
'=============================================================

 MENULVL = 1

 IF SIMNUM = 1 THEN
 Simulation = 1
 CALL SIM1
 ELSEIF SIMNUM = 2 THEN
 Simulation = 2
 CALL SIM1
 ELSEIF SIMNUM = 3 THEN
 CALL SAVEDATA
 ELSEIF SIMNUM = 4 THEN
 CALL LOADATA
 ELSEIF SIMNUM = 5 THEN
 CALL RUNDISPLAY
 ELSEIF SIMNUM = 0 THEN
 DONE = 1
 END IF

LOOP UNTIL DONE = 1

CALL WRITESTAT
SCREEN 0
COLOR 15, 0 'White on Black
CLS

END

'
'=============================================================
' Check for what error has occurred
'=============================================================
```

```
ErrorHandler:

SCREEN 0
SCREEN 12
PRINT " "
PRINT " "
PRINT " "
PRINT " "
PRINT " "
PRINT " An Error has occured - Press any key to continue."
DO
LOOP WHILE INKEY$ = ""

RUN CSSMENU$


SUB CLEARINFOBOX

'-----------------------------------------------------------

'===========================================================
' Program Name: CLEARINFOBOX
' Description: This subroutine clears the message at the
' bottom of the screen.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'===========================================================
'

'$INCLUDE: 'MENUPAR.INC'

COLOR WHITE, CYAN

LCOL% = 4
TROW% = 21
BROW% = 24
RCOL% = 76
LABEL$ = ""
FORE% = CYAN
BACK% = CYAN
PAGE% = 0
FRAME% = 1
TYP% = 0

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

END SUB

SUB DISPDYNMODEL

'-----------------------------------------------------------

'===========================================================
' Program Name: DISPDYNMODEL
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'===========================================================
'

'$INCLUDE: 'MENUPAR.INC'

HIGHLT = YELLOW
```

```
LCOL% = 18
TROW% = 3
BROW% = 11
RCOL% = 65
LABEL$ = "Dynamic Model Menu"
FORE% = WHITE
BACK% = CYAN
PAGE% = 0
FRAME% = 1

IF MENULVL = 2 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 4
COLM = 33

COLOR YELLOW, CYAN

LOCATE ROW, COLM - 14
PRINT "Press the Number to Select a Model"

COLOR WHITE, CYAN

LOCATE ROW + 2, COLM
PRINT "1. DLVO Theory Model"

LOCATE ROW + 3, COLM
PRINT "2. Acid/Base Model"

LOCATE ROW + 4, COLM
PRINT "3. Random Model"

LOCATE ROW + 5, COLM
PRINT "3. Time Increment:"
LOCATE ROW + 5, COLM + 29
PRINT "sec."

LOCATE ROW + 7, COLM - 9
PRINT "Help "

LOCATE ROW + 7, COLM + 1
PRINT "<ESC> To Return"

COLOR YELLOW, CYAN

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
PRINT "4."

LOCATE ROW + 7, COLM - 9
PRINT "H"

LOCATE ROW + 7, COLM + 1
PRINT "<ESC>"

END SUB


SUB DISPELECTRO1

'-------------------------------------------------------

'=======================================================
```

```
' Program Name: DISPELECTRO1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

'$INCLUDE: 'MENUPAR.INC'

HIGHLT = YELLOW

LCOL% = 24
TROW% = 4
BROW% = 11
RCOL% = 70
LABEL$ = "Electrolyte Information Menu"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1

IF MENULVL = 3 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 5
COLM = 25

COLOR YELLOW, BLUE

LOCATE ROW, COLM
PRINT "Press the Number of the Option to be Modified"

COLOR WHITE, BLUE

LOCATE ROW + 2, COLM
PRINT "1. Common Name: "

LOCATE ROW + 3, COLM
PRINT "2. Chemical Name: "

LOCATE ROW + 4, COLM
PRINT "3. Concentration: "
LOCATE ROW + 4, COLM + 30
PRINT "Normality"

LOCATE ROW + 6, COLM + 7
PRINT "Help "

LOCATE ROW + 6, COLM + 17
PRINT "<ESC> To Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 6, COLM + 7
PRINT "H"
```

```
LOCATE ROW + 6, COLM + 17
PRINT "<ESC>"

END SUB

SUB DISPERR (MESSAGE$())

'-------------------------------------------------------

'=======================================================
' Program Name: DISPERR
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

'$INCLUDE: 'MENUPAR.INC'

SCREEN 0, , 2
COLOR WHITE, CYAN
CLS

LCOL% = 5
TROW% = 2
BROW% = 23
RCOL% = 78
LABEL$ = "Error Menu"
FORE% = WHITE
BACK% = BLUE
PAGE% = 2
FRAME% = 1
TYP% = 3

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

COLOR YELLOW, BLUE

FOR M = 1 TO 19
 LOCATE M + 2, 8
 PRINT MESSAGE$(M)
NEXT M

 LOCATE 23, 27
 COLOR WHITE, BLUE
 PRINT "Hit Any Key to Continue ..."
 DO
 LOOP WHILE INKEY$ = ""

SCREEN 0, , 0
END SUB

DEFSNG A-Z
SUB DISPFIBER

'-------------------------------------------------------

'=======================================================
' Program Name: DISPFIBER
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
```

```
'
'$INCLUDE: 'MENUPAR.INC'

HIGHLT = YELLOW

LCOL% = 24
TROW% = 4
BROW% = 15
RCOL% = 70
LABEL$ = "Fiber Information Menu"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1

IF MENULVL = 3 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 5
COLM = 25

COLOR YELLOW, BLUE

LOCATE ROW, COLM
PRINT "Press the Number of the Option to be Modified"

COLOR WHITE, BLUE

LOCATE ROW + 2, COLM
PRINT "1. Common Name: "

LOCATE ROW + 3, COLM
PRINT "2. Chemical Name: "

LOCATE ROW + 4, COLM
PRINT "3. Shape: "

LOCATE ROW + 5, COLM
PRINT "4. Diameter: "
LOCATE ROW + 5, COLM + 20
PRINT "um"

LOCATE ROW + 6, COLM
PRINT "5. Density: "
LOCATE ROW + 6, COLM + 21
PRINT "gm/cm^3"

LOCATE ROW + 7, COLM
PRINT "6. Hamaker Constant: "
LOCATE ROW + 7, COLM + 34
PRINT "Joules"

LOCATE ROW + 9, COLM + 8
PRINT "Help"

LOCATE ROW + 9, COLM + 18
PRINT "<ESC> to Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
```

```
 PRINT "4."

 LOCATE ROW + 6, COLM
 PRINT "5."

LOCATE ROW + 7, COLM
PRINT "6."

LOCATE ROW + 9, COLM + 8
PRINT "H"

LOCATE ROW + 9, COLM + 18
PRINT "<ESC>"

END SUB

DEFINT A-Z
SUB DISPHELP (HELPMES$())

'-----------------------------------------------------

'=====================================================
' Program Name: DISPHELP
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'

'$INCLUDE: 'MENUPAR.INC'

SCREEN 0, , 2

COLOR WHITE, CYAN
CLS

LCOL% = 5
TROW% = 2
BROW% = 23
RCOL% = 78
LABEL$ = "Help Menu"
FORE% = WHITE
BACK% = BLUE
PAGE% = 2
FRAME% = 1
TYP% = 3

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

COLOR YELLOW, BLUE

FOR M = 1 TO 19
 LOCATE M + 2, 8
 PRINT HELPMES$(M)
NEXT M

 LOCATE 23, 27
 COLOR WHITE, BLUE
 PRINT "Hit Any Key to Continue ..."
 DO
 LOOP WHILE INKEY$ = ""


END SUB

SUB DISPINFOBOX

'-----------------------------------------------------

'=====================================================
' Program Name: DISPINFOBOX
```

```
'
' Description: This subroutine displays a message at the
' bottom of the screen.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

'$INCLUDE: 'MENUPAR.INC'

COLOR WHITE, CYAN

LCOL% = 4
TROW% = 21
BROW% = 24
RCOL% = 76
LABEL$ = ""
FORE% = WHITE
BACK% = CYAN
PAGE% = 0
FRAME% = 1
TYP% = 0

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)


END SUB

DEFSNG A-Z
SUB DISPMAIN

'-------------------------------------------------------------
'=============================================================
' Program Name: DISPMAIN
'
' Description: This subroutine displays the main menu.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'
'$INCLUDE: 'MENUPAR.INC'

FORGND = WHITE
BACGND = BLUE
FORGND2 = WHITE
BACGND2 = CYAN

HIGHLT = YELLOW

SCREEN 0

COLOR WHITE, CYAN
CLS

LCOL% = 5
TROW% = 2
BROW% = 17
RCOL% = 78
LABEL$ = "Colloidal Suspension Simulator"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1
IF MENULVL = 0 THEN
```

```
 TYP% = 3
ELSE
 TYP% = 2
END IF

'CLS

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

LCOL% = 6
TROW% = 4
BROW% = 13
RCOL% = 34
LABEL$ = "Main Menu"
FORE% = 15
BACK% = 3
PAGE% = 0
FRAME% = 1
IF MENULVL = 0 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 5
COLM = 7

COLOR YELLOW, CYAN

LOCATE ROW, COLM + 3
PRINT "Enter Selection"

COLOR WHITE, CYAN

LOCATE ROW + 2, COLM
PRINT "1. Particle A - Particle B"

LOCATE ROW + 3, COLM
PRINT "2. Particle A - Fiber"

LOCATE ROW + 4, COLM
PRINT "3. Save Configuration"

LOCATE ROW + 5, COLM
PRINT "4. Load Configuration"

LOCATE ROW + 6, COLM
PRINT "D. Display the Model"

LOCATE ROW + 8, COLM + 2
PRINT "Help Exit"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
PRINT "4."

LOCATE ROW + 6, COLM
PRINT "D."

LOCATE ROW + 8, COLM + 2
PRINT "H"

LOCATE ROW + 8, COLM + 15
PRINT "x"
```

```
END SUB

SUB DISPMATERIAL1

'-------------------------------------------------------

'=======================================================
' Program Name: DISPMATERIAL1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

'$INCLUDE: 'MENUPAR.INC'

SCREEN 0

HIGHLT = YELLOW

LCOL% = 18
TROW% = 3
BROW% = 11
RCOL% = 57
LABEL$ = "Material Parameters Menu"
FORE% = WHITE
BACK% = CYAN
PAGE% = 0
FRAME% = 1

IF MENULVL = 2 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 4
COLM = 19

COLOR YELLOW, CYAN

LOCATE ROW, COLM + 3
PRINT "Enter Selection"

COLOR WHITE, CYAN

LOCATE ROW + 2, COLM
PRINT "1. Modify Particle A Information."

IF Simulation = 1 THEN
 LOCATE ROW + 3, COLM
 PRINT "2. Modify Particle B Information."
ELSEIF Simulation = 2 THEN
 LOCATE ROW + 3, COLM
 PRINT "2. Modify Fiber Information."
END IF

LOCATE ROW + 4, COLM
PRINT "3. Modify the Suspension Medium Info."

LOCATE ROW + 5, COLM
PRINT "4. Modify the Electrolyte Info."

LOCATE ROW + 7, COLM + 5
PRINT "Help "

LOCATE ROW + 7, COLM + 15
```

```
PRINT "<ESC> To Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
PRINT "4."

LOCATE ROW + 7, COLM + 5
PRINT "H"

LOCATE ROW + 7, COLM + 15
PRINT "<ESC>"

END SUB

DEFINT A-Z
SUB DISPMEDIUM1

'-----------------------------------------------------------

'===========================================================
' Program Name: DISPMEDIUM1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'===========================================================
'

'$INCLUDE: 'MENUPAR.INC'

HIGHLT = YELLOW

LCOL% = 24
TROW% = 4
BROW% = 15
RCOL% = 70
LABEL$ = "Medium Information Menu"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1

IF MENULVL = 3 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 5
COLM = 25

COLOR YELLOW, BLUE

LOCATE ROW, COLM
PRINT "Press the Number of the Option to be Modified"

COLOR WHITE, BLUE

LOCATE ROW + 2, COLM
```

```
PRINT "1. Common Name: "

LOCATE ROW + 3, COLM
PRINT "2. Chemical Name: "

LOCATE ROW + 4, COLM
PRINT "3. Density: "
LOCATE ROW + 4, COLM + 21
PRINT "gm/cm^3"

LOCATE ROW + 5, COLM
PRINT "4. Viscosity: "
LOCATE ROW + 5, COLM + 23
PRINT "gm/(cm sec)"

LOCATE ROW + 6, COLM
PRINT "5. Hamaker Constant: "
LOCATE ROW + 6, COLM + 33
PRINT "Joules"

LOCATE ROW + 7, COLM
PRINT "6. Permeability: "

LOCATE ROW + 8, COLM
PRINT "7. Temperature: "

LOCATE ROW + 8, COLM + 24
PRINT " Degrees Celesius "

LOCATE ROW + 10, COLM + 8
PRINT "Help "

LOCATE ROW + 10, COLM + 18
PRINT "<ESC> To Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
PRINT "4."

LOCATE ROW + 6, COLM
PRINT "5."

LOCATE ROW + 7, COLM
PRINT "6."

LOCATE ROW + 8, COLM
PRINT "7."

LOCATE ROW + 10, COLM + 8
PRINT "H"

LOCATE ROW + 10, COLM + 18
PRINT "<ESC>"

END SUB

DEFSNG A-Z
SUB DISPPARTA

'----------------------------------------------------------

'==========================================================
' Program Name: DISPPARTA
'
' Description: This subroutine displays the menu for the
' particle information menu.
'
' Author: Peter T. Robinson
```

```
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

'$INCLUDE: 'MENUPAR.INC'

SCREEN 0

HIGHLT = YELLOW

LCOL% = 24
TROW% = 4
BROW% = 17
RCOL% = 70
LABEL$ = "Particle A Information Menu"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1

IF MENULVL = 3 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 5
COLM = 25

COLOR YELLOW, BLUE

LOCATE ROW, COLM
PRINT "Press the Number of the Option to be Modified"

COLOR WHITE, BLUE

LOCATE ROW + 2, COLM
PRINT "1. Common Name: "

LOCATE ROW + 3, COLM
PRINT "2. Chemical Name: "

LOCATE ROW + 4, COLM
PRINT "3. Shape: "

LOCATE ROW + 5, COLM
PRINT "4. Average Initial Velocity: "
LOCATE ROW + 5, COLM + 36
PRINT "um/sec"

LOCATE ROW + 6, COLM
PRINT "5. Mean Diameter: "
LOCATE ROW + 6, COLM + 25
PRINT "um"

LOCATE ROW + 7, COLM
PRINT "6. Diameter Standard Dev.: "

LOCATE ROW + 8, COLM
PRINT "7. Density: "
LOCATE ROW + 8, COLM + 21
PRINT "gm/cm^3"

LOCATE ROW + 9, COLM
PRINT "8. Number of Particles: "

LOCATE ROW + 10, COLM
PRINT "9. Hamaker Constant: "
LOCATE ROW + 10, COLM + 34
PRINT "Joules"
```

```
LOCATE ROW + 12, COLM + 7
PRINT "Help "

LOCATE ROW + 12, COLM + 17
PRINT "<ESC> To Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
PRINT "4."

LOCATE ROW + 6, COLM
PRINT "5."

LOCATE ROW + 7, COLM
PRINT "6."

LOCATE ROW + 8, COLM
PRINT "7."

LOCATE ROW + 9, COLM
PRINT "8."

LOCATE ROW + 10, COLM
PRINT "9."

LOCATE ROW + 12, COLM + 7
PRINT "H"

LOCATE ROW + 12, COLM + 17
PRINT "<ESC>"

END SUB

DEFINT A-Z
SUB DISPPARTB

'-------------------------------------------------------

'=======================================================
' Program Name: DISPPARTB
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

'$INCLUDE: 'MENUPAR.INC'

HIGHLT = YELLOW

LCOL% = 24
TROW% = 4
BROW% = 17
RCOL% = 70
LABEL$ = "Particle B Information Menu"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1
```

```
IF MENULVL = 3 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 5
COLM = 25

COLOR YELLOW, BLUE

LOCATE ROW, COLM
PRINT "Press the Number of the Option to be Modified"

COLOR WHITE, BLUE

LOCATE ROW + 2, COLM
PRINT "1. Common Name: "

LOCATE ROW + 3, COLM
PRINT "2. Chemical Name: "


LOCATE ROW + 4, COLM
PRINT "3. Shape: "

LOCATE ROW + 5, COLM
PRINT "4. Average Initial Velocity: "
LOCATE ROW + 5, COLM + 36
PRINT "um/sec"

LOCATE ROW + 6, COLM
PRINT "5. Mean Diameter: "
LOCATE ROW + 6, COLM + 25
PRINT "um"

LOCATE ROW + 7, COLM
PRINT "6. Diameter Standard Dev.: "

LOCATE ROW + 8, COLM
PRINT "7. Density: "
LOCATE ROW + 8, COLM + 21
PRINT "gm/cm^3"

LOCATE ROW + 9, COLM
PRINT "8. Number of Particles: "

LOCATE ROW + 10, COLM
PRINT "9. Hamaker Constant: "
LOCATE ROW + 10, COLM + 34
PRINT "Joules"

LOCATE ROW + 12, COLM + 8
PRINT "Help "

LOCATE ROW + 12, COLM + 18
PRINT "<ESC> to Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
PRINT "4."

LOCATE ROW + 6, COLM
PRINT "5."
```

```
LOCATE ROW + 7, COLM
PRINT "6."

LOCATE ROW + 8, COLM
PRINT "7."

LOCATE ROW + 9, COLM
PRINT "8."

LOCATE ROW + 10, COLM
PRINT "9."

LOCATE ROW + 12, COLM + 8
PRINT "H"

LOCATE ROW + 12, COLM + 18
PRINT "<ESC>"

END SUB

DEFSNG A-Z
SUB DISPSIM1

'------------------------------------------------------------

'============================================================
' Program Name: DISPSIM1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'============================================================
'
'$INCLUDE: 'MENUPAR.INC'

SCREEN 0

HIGHLT = YELLOW

LCOL% = 12
TROW% = 6
BROW% = 15
RCOL% = 47

IF Simulation = 1 THEN
 LABEL$ = "Particle A - Particle B Menu"
ELSEIF Simulation = 2 THEN
 LABEL$ = "Particle A - Fiber Menu"
END IF

FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1
IF MENULVL = 1 THEN
 TYP% = 3
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 7
COLM = 13

COLOR YELLOW, BLUE

LOCATE ROW, COLM + 3
PRINT "Enter Selection"
```

```
COLOR WHITE, BLUE
LOCATE ROW + 2, COLM
PRINT "1. Modify the Material Parameters."

LOCATE ROW + 3, COLM
PRINT "2. Modify the Surface Parameters."

LOCATE ROW + 4, COLM
PRINT "3. Choose A Dynamic Model."

LOCATE ROW + 5, COLM
PRINT "R. Run the Model"

LOCATE ROW + 7, COLM + 5
PRINT "Help"

LOCATE ROW + 7, COLM + 15
PRINT "<ESC> To Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 5, COLM
PRINT "R."

LOCATE ROW + 7, COLM + 5
PRINT "H"

LOCATE ROW + 7, COLM + 15
PRINT "<ESC>"

END SUB

DEFINT A-Z
SUB DISPSURFACE1

'-------------------------------------------------------

'=======================================================
' Program Name: DISPSURFACE1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

'$INCLUDE: 'MENUPAR.INC'

HIGHLT = YELLOW

LCOL% = 18
TROW% = 3
BROW% = 10
RCOL% = 70
LABEL$ = "Surface Parameter Menu"
FORE% = WHITE
BACK% = CYAN
PAGE% = 0
FRAME% = 1

IF MENULVL = 2 THEN
 TYP% = 3
```

```
ELSE
 TYP% = 2
END IF

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

ROW = 4
COLM = 19

COLOR YELLOW, CYAN

LOCATE ROW, COLM
PRINT "Press the Number of the Option to be Modified"

COLOR WHITE, CYAN

LOCATE ROW + 2, COLM
PRINT "1. pH of the Medium: "

LOCATE ROW + 3, COLM
PRINT "2. Zeta Potential of Particle A: "
LOCATE ROW + 3, COLM + 42
PRINT "meV"

IF Simulation = 1 THEN
 LOCATE ROW + 4, COLM
 PRINT "3. Zeta Potential of Particle B: "
 LOCATE ROW + 4, COLM + 42
 PRINT "meV"

ELSEIF Simulation = 2 THEN
 LOCATE ROW + 4, COLM
 PRINT "3. Zeta Potential of the Fiber: "
 LOCATE ROW + 4, COLM + 42
 PRINT "meV"

END IF

LOCATE ROW + 6, COLM + 7
PRINT "Help "

LOCATE ROW + 6, COLM + 17
PRINT "<ESC> To Return"

COLOR HIGHLT

LOCATE ROW + 2, COLM
PRINT "1."

LOCATE ROW + 3, COLM
PRINT "2."

LOCATE ROW + 4, COLM
PRINT "3."

LOCATE ROW + 6, COLM + 7
PRINT "H"

LOCATE ROW + 6, COLM + 17
PRINT "<ESC>"

END SUB

SUB DYNMODEL

'-------------------------------------------------------

'=======================================================
' Program Name: DYNMODEL
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
```

```
' Revision History:
' None
'
'=============================================================
'
DONE = 0

DO
'=================================
' Display the Particle Information Menu for Model One.
'=================================

IF MENULVL = 2 THEN
CALL DISPDYNMODEL
ELSE
CALL DISPMAIN
CALL DISPSIM1
CALL DISPDYNMODEL
END IF

'=================================
' Accept and verify the input to the Particle Information Menu.
'=================================

CALL GETDYNMODEL(OPT%)

IF OPT% = 0 THEN ' Return to the previous menu
DONE = 1
END IF

LOOP UNTIL DONE = 1


END SUB

SUB ELECTRO1

'-------------------------------------------------------------
'=============================================================
' Program Name: ELECTRO1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

DONE = 0

DO
'=================================
' Display the Particle Information Menu for Model One.
'=================================

IF MENULVL = 3 THEN
CALL DISPELECTRO1
ELSE
CALL DISPMAIN
CALL DISPSIM1
CALL DISPMATERIAL1
CALL DISPELECTRO1

END IF

'=================================
' Accept and verify the input to the Particle Information Menu.
'=================================

CALL GETELECTRO1(OPT%)

IF OPT% = 0 THEN ' Return to the previous menu
```

```
DONE = 1
END IF

LOOP UNTIL DONE = 1

END SUB

DEFSNG A-Z
SUB FIBER

'------------------------------------------------------

'=======================================================
' Program Name: FIBER
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

DONE = 0

DO
'=================================
' Display the Fiber Information Menu for Model One.
'=================================

IF MENULVL = 3 THEN
CALL DISPFIBER
ELSE
CALL DISPMAIN
CALL DISPSIM1
CALL DISPMATERIAL1
CALL DISPFIBER

END IF

'=================================
' Accept and verify the input to the Particle Information Menu.
'=================================

CALL GETFIBER(OPT%)

IF OPT% = 0 THEN ' Return to the previous menu
DONE = 1
END IF

LOOP UNTIL DONE = 1

END SUB

DEFINT A-Z
SUB GETDYNMODEL (OPT%)

'------------------------------------------------------

'=======================================================
' Program Name: GETDYNMODEL
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'
```

```
'$INCLUDE: 'MENUPAR.INC'
OPT% = 0
DONE = 0
QUIT = 0

ROW = 6
COLM = 19

LOCATE ROW + 3, COLM + 32
PRINT USING "##.###^^^^"; TimeIncr
'PRINT USING "##.######"; TimeIncr


 FORGND1 = CYAN
 FORGND2 = CYAN
 FORGND3 = CYAN

IF DYNMDL = 1 THEN
 FORGND1 = YELLOW
ELSEIF DYNMDL = 2 THEN
 FORGND2 = YELLOW
ELSEIF DYNMDL = 3 THEN
 FORGND3 = YELLOW
END IF

COLOR FORGND1, CYAN
LOCATE ROW, COLM
PRINT "Selected --> "

COLOR FORGND2, CYAN
LOCATE ROW + 1, COLM
PRINT "Selected --> "

COLOR FORGND3, CYAN
LOCATE ROW + 2, COLM
PRINT "Selected --> "

DO
CHOICE$ = INKEY$

IF CHOICE$ = "1" OR CHOICE$ = "2" OR CHOICE$ = "3" THEN
VALID = 0

FORGND1 = CYAN
FORGND2 = CYAN
FORGND3 = CYAN

IF CHOICE$ = "1" THEN
FORGND1 = YELLOW
DYNMDL = 1
ELSEIF CHOICE$ = "2" THEN
FORGND2 = YELLOW
DYNMDL = 2
ELSEIF CHOICE$ = "3" THEN
FORGND3 = YELLOW
DYNMDL = 3
END IF

COLOR FORGND1, CYAN
LOCATE ROW, COLM
PRINT "Selected --> "

COLOR FORGND2, CYAN
LOCATE ROW + 1, COLM
PRINT "Selected --> "

COLOR FORGND3, CYAN
LOCATE ROW + 2, COLM
PRINT "Selected --> "

ELSEIF CHOICE$ = "4" THEN

VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
```

```
 COLOR YELLOW, CYAN
 LOCATE 21, 20
 PRINT "Time Increment: ";
 PRINT USING "##.###^^^^"; TimeIncr
' PRINT USING "##.######"; TimeIncr
COLOR WHITE, CYAN
LOCATE 22, 25
PRINT "Change to: "
LOCATE 22, 36
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 36
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 99.999 THEN
TimeIncr = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 99.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 3, COLM + 32
COLOR YELLOW, CYAN
PRINT USING "##.###^^^^"; TimeIncr
' PRINT USING "##.######"; TimeIncr
CALL CLEARINFOBOX

ELSEIF CHOICE$ = CHR$(27) THEN 'Escape
OPT% = 0
DONE = 1
ELSEIF UCASE$(CHOICE$) = "H" THEN
CALL HELPDYN
SCREEN 0, , 0
END IF
LOOP UNTIL DONE = 1

IF QUIT = 1 THEN
 CLS
 END
END IF

END SUB

SUB GETELECTRO1 (OPT%)

'-------------------------------------------------------

'=======================================================
' Program Name: GETELECTRO1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

'$INCLUDE: 'MENUPAR.INC'

OPT% = 0
DONE = 0
QUIT = 0
```

```
ROW = 7
COLM = 40

COLOR YELLOW, BLUE

LOCATE ROW, COLM + 2
PRINT ElectroInfo1.ComName

LOCATE ROW + 1, COLM + 4
PRINT ElectroInfo1.ChemName

LOCATE ROW + 2, COLM + 3
PRINT USING "##.########"; ElectroInfo1.Concen

DO
 CHOICE$ = INKEY$

 IF CHOICE$ = "1" THEN
 CALL DISPINFOBOX
 BUFFER$ = ""
 LOCATE 21, 18
 COLOR YELLOW, CYAN
 PRINT "Common Name: "; ElectroInfo1.ComName
 LOCATE 22, 20
 COLOR WHITE, CYAN
 PRINT "Change to: "
 LOCATE 22, 31
 COLOR YELLOW, CYAN
 INPUT ; "", BUFFER$
 IF BUFFER$ <> "" THEN ElectroInfo1.ComName = BUFFER$
 LOCATE ROW, COLM + 2
 COLOR YELLOW, BLUE
 PRINT ElectroInfo1.ComName
 CALL CLEARINFOBOX

 ELSEIF CHOICE$ = "2" THEN

 CALL DISPINFOBOX
 BUFFER$ = ""
 COLOR YELLOW, CYAN
 LOCATE 21, 18
 PRINT "Chemical Name: "; ElectroInfo1.ChemName
 COLOR WHITE, CYAN
 LOCATE 22, 22
 PRINT "Change to: "
 LOCATE 22, 33
 INPUT ; "", BUFFER$
 IF BUFFER$ <> "" THEN ElectroInfo1.ChemName = BUFFER$
 LOCATE ROW + 1, COLM + 3
 COLOR YELLOW, BLUE
 PRINT ElectroInfo1.ChemName
 CALL CLEARINFOBOX

 ELSEIF CHOICE$ = "3" THEN
 VALID = 0
 CALL DISPINFOBOX
 TEMPBUF$ = ""
 DO
 BUFFER$ = ""
 COLOR YELLOW, CYAN
 LOCATE 21, 20
 PRINT "Concentration:";
 PRINT USING "###.########"; ElectroInfo1.Concen
 COLOR WHITE, CYAN
 LOCATE 22, 24
 PRINT "Change to: "
 LOCATE 22, 36
 COLOR CYAN, CYAN
 PRINT TEMPBUF$
 COLOR WHITE, CYAN
 LOCATE 22, 36
 INPUT ; "", BUFFER$
 TEMPBUF$ = BUFFER$
 IF BUFFER$ <> "" THEN

 BUFFER! = VAL(BUFFER$)
 IF BUFFER! > 0 AND BUFFER! <= 1! THEN
 ElectroInfo1.Concen = BUFFER!
```

```
 VALID = 1
 ELSE
 LOCATE 23, 16
 COLOR YELLOW, CYAN
 PRINT "Valid input is a real number between 0 and 1.0"
 END IF
 ELSE
 VALID = 1
 END IF
 LOOP UNTIL VALID = 1

 LOCATE ROW + 2, COLM + 3
 COLOR YELLOW, BLUE
 PRINT USING "##.########"; ElectroInfo1.Concen
 CALL CLEARINFOBOX

 ELSEIF CHOICE$ = CHR$(27) THEN 'Escape
 OPT% = 0
 DONE = 1
 ELSEIF UCASE$(CHOICE$) = "H" THEN
 CALL HELPELECTRO
 SCREEN 0, , 0
 END IF
LOOP UNTIL DONE = 1

IF QUIT = 1 THEN
 CLS
 END
END IF

END SUB

DEFSNG A-Z
SUB GETFIBER (OPT%)

'-----------------------------------------------------------

'===========================================================
' Program Name: GETFIBER
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'===========================================================
'

'$INCLUDE: 'MENUPAR.INC'

OPT% = 0
DONE = 0
QUIT = 0

ROW = 7
COLM = 40

COLOR YELLOW, BLUE

LOCATE ROW, COLM + 2
PRINT FiberInfo.ComName

LOCATE ROW + 1, COLM + 4
PRINT FiberInfo.ChemName

LOCATE ROW + 2, COLM - 4
PRINT FiberInfo.Shape

LOCATE ROW + 3, COLM - 2
PRINT USING "##.###"; FiberInfo.Diameter

LOCATE ROW + 4, COLM - 3
PRINT USING "###.###"; FiberInfo.Density
```

```
LOCATE ROW + 5, COLM + 6
PRINT USING "###.###^^^^"; FiberInfo.Hamaker

BLANK$ = " "
DO
CHOICE$ = INKEY$

IF CHOICE$ = "1" THEN
CALL DISPINFOBOX
BUFFER$ = ""
LOCATE 21, 18
COLOR YELLOW, CYAN
PRINT "Common Name: "; FiberInfo.ComName
LOCATE 22, 20
COLOR WHITE, CYAN
PRINT "Change to: "
LOCATE 22, 31
COLOR YELLOW, CYAN
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN FiberInfo.ComName = BUFFER$
LOCATE ROW, COLM + 2
COLOR YELLOW, BLUE
PRINT FiberInfo.ComName
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "2" THEN

CALL DISPINFOBOX
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 18
PRINT "Chemical Name: "; FiberInfo.ChemName
COLOR WHITE, CYAN
LOCATE 22, 22
PRINT "Change to: "
LOCATE 22, 33
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN FiberInfo.ChemName = BUFFER$
LOCATE ROW + 1, COLM + 4
COLOR YELLOW, BLUE
PRINT FiberInfo.ChemName
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "3" THEN

CALL DISPINFOBOX
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 24
PRINT "Shape: "; FiberInfo.Shape
COLOR WHITE, CYAN
LOCATE 22, 20
PRINT "Change to: "
LOCATE 22, 31
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN FiberInfo.Shape = BUFFER$
LOCATE ROW + 2, COLM - 4
COLOR YELLOW, BLUE
PRINT FiberInfo.Shape
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "4" THEN
CALL DISPINFOBOX
VALID = 0
TEMPBUF$ = ""
DO
COLOR YELLOW, CYAN
LOCATE 21, 22
PRINT "Diameter: ";
PRINT USING "##.###"; FiberInfo.Diameter
COLOR WHITE, CYAN
LOCATE 22, 21
PRINT "Change to: "
LOCATE 22, 32
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
```

```
LOCATE 22, 32
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$

IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 99.999 THEN
FiberInfo.Diameter = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 99.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 3, COLM - 2
COLOR YELLOW, BLUE
PRINT USING "##.###"; FiberInfo.Diameter
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "5" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 21
PRINT "Density:";
PRINT USING "###.###"; FiberInfoF.Density
COLOR WHITE, CYAN
LOCATE 22, 19
PRINT "Change to: "
LOCATE 22, 30
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 30
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 999.999 THEN
FiberInfo.Density = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 4, COLM - 3
COLOR YELLOW, BLUE
PRINT USING "###.###"; FiberInfo.Density
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "6" THEN

VALID = 0
TEMPBUF$ = ""
CALL DISPINFOBOX

DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 16
PRINT "Hamaker Constant:";
PRINT USING "###.###^^^^"; FiberInfo.Hamaker
```

```
LOCATE 22, 23
COLOR WHITE, CYAN
PRINT "Change to: "
COLOR CYAN, CYAN
LOCATE 22, 37
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 34
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 1E+30 THEN
FiberInfo.Hamaker = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is an integer number between 1 and 1.0E+30"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 5, COLM + 6
COLOR YELLOW, BLUE
PRINT USING "###.###^^^^"; FiberInfo.Hamaker
CALL CLEARINFOBOX

ELSEIF UCASE$(CHOICE$) = "H" THEN
CALL HELPFIBER
SCREEN 0, , 0

ELSEIF CHOICE$ = CHR$(27) THEN 'Escape
OPT% = 0
DONE = 1
END IF

LOOP UNTIL DONE = 1

IF QUIT = 1 THEN
 CLS
 END
END IF

END SUB

DEFINT A-Z
SUB GETFILENAM (FILENAM$)

'------------------------------------------------------------

'============================================================
' Program Name: GETFILENAM
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'============================================================
'
'$INCLUDE: 'MENUPAR.INC'

 CALL DISPINFOBOX
 BUFFER$ = ""
 LOCATE 22, 5
 COLOR YELLOW, CYAN
 PRINT "Enter Filename (With no File Extension): ";
 COLOR WHITE, CYAN
 LOCATE 22, 47
 INPUT ; "", BUFFER$
 FILENAM$ = BUFFER$
```

```
CALL CLEARINFOBOX

END SUB

SUB GETMAIN (SIMNUM)

'-------------------------------------------------------

'=======================================================
' Program Name: GETMAIN
'
' Description: This routine gets the user responses from the
' DISPMAIN menu.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

QUIT = 0
SIMNUM = 0
DONE = 0

DO
 CHOICE$ = INKEY$

 IF CHOICE$ = "1" THEN
 SIMNUM = 1
 DONE = 1
 NUMPART = 60
 IF PartInfoA.Number > NUMPART THEN PartInfoA.Number = NUMPART
 ELSEIF CHOICE$ = "2" THEN
 SIMNUM = 2
 DONE = 1
 ELSEIF CHOICE$ = "3" THEN
 SIMNUM = 3
 DONE = 1
 ELSEIF CHOICE$ = "4" THEN
 SIMNUM = 4
 DONE = 1
 ELSEIF UCASE$(CHOICE$) = "D" THEN
 SIMNUM = 5
 DONE = 1
 ELSEIF UCASE$(CHOICE$) = "H" THEN
 CALL HELPMAIN
 SCREEN 0, , 0
 ELSEIF UCASE$(CHOICE$) = "X" THEN
 SIMNUM = 0
 QUIT = 1
 DONE = 1
 END IF
LOOP UNTIL DONE = 1

END SUB

DEFSNG A-Z
SUB GETMATERIAL1 (OPT%)

'-------------------------------------------------------

'=======================================================
' Program Name: GETMATERIAL1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
```

```
'

OPT% = 0
DONE = 0
QUIT = 0

DO
 CHOICE$ = INKEY$

 IF CHOICE$ = "1" THEN
 OPT% = 1
 DONE = 1
 ELSEIF CHOICE$ = "2" THEN
 OPT% = 2
 DONE = 1
 ELSEIF CHOICE$ = "3" THEN
 OPT% = 3
 DONE = 1
 ELSEIF CHOICE$ = "4" THEN
 OPT% = 4
 DONE = 1
 ELSEIF UCASE$(CHOICE$) = "H" THEN
 CALL HELPMATRL
 SCREEN 0, , 0
 ELSEIF CHOICE$ = CHR$(27) THEN
 OPT% = 0
 DONE = 1
 END IF
LOOP WHILE DONE = 0

IF QUIT = 1 THEN
 CLS
 END
END IF

END SUB

DEFINT A-Z
SUB GETMEDIUM1 (OPT%)
```

```
'-----------------------------------------------

'===============================================
' Program Name: GETMEDIUM1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'===============================================
'

'$INCLUDE: 'MENUPAR.INC'

OPT% = 0
DONE = 0
QUIT = 0

ROW = 7
COLM = 40

COLOR YELLOW, BLUE

LOCATE ROW, COLM + 2
PRINT MediumInfo1.ComName

LOCATE ROW + 1, COLM + 4
PRINT MediumInfo1.ChemName

LOCATE ROW + 2, COLM - 3
PRINT USING "###.###"; MediumInfo1.Density

LOCATE ROW + 3, COLM - 1
```

```
PRINT USING "###.###"; MediumInfo1.Viscosity

LOCATE ROW + 4, COLM + 6
PRINT USING "###.###^^^^"; MediumInfo1.Hamaker

LOCATE ROW + 5, COLM + 2
PRINT USING "####.###"; MediumInfo1.Perm

LOCATE ROW + 6, COLM
PRINT USING "#####.###"; MediumInfo1.Temp

DO
 CHOICE$ = INKEY$

 IF CHOICE$ = "1" THEN
 CALL DISPINFOBOX
 BUFFER$ = ""
 LOCATE 21, 18
 COLOR YELLOW, CYAN
 PRINT "Common Name: "; MediumInfo1.ComName
 LOCATE 22, 20
 COLOR WHITE, CYAN
 PRINT "Change to: "
 LOCATE 22, 31
 COLOR YELLOW, CYAN
 INPUT ; "", BUFFER$
 IF BUFFER$ <> "" THEN MediumInfo1.ComName = BUFFER$
 LOCATE ROW, COLM + 2
 COLOR YELLOW, BLUE
 PRINT MediumInfo1.ComName
 CALL CLEARINFOBOX

 ELSEIF CHOICE$ = "2" THEN

 CALL DISPINFOBOX
 BUFFER$ = ""
 COLOR YELLOW, CYAN
 LOCATE 21, 18
 PRINT "Chemical Name: "; MediumInfo1.ChemName
 COLOR WHITE, CYAN
 LOCATE 22, 22
 PRINT "Change to: "
 LOCATE 22, 33
 INPUT ; "", BUFFER$
 IF BUFFER$ <> "" THEN MediumInfo1.ChemName = BUFFER$
 LOCATE ROW + 1, COLM + 3
 COLOR YELLOW, BLUE
 PRINT MediumInfo1.ChemName
 CALL CLEARINFOBOX

 ELSEIF CHOICE$ = "3" THEN
 VALID = 0
 CALL DISPINFOBOX
 TEMPBUF$ = ""
 DO
 BUFFER$ = ""
 COLOR YELLOW, CYAN
 LOCATE 21, 21
 PRINT "Density: ";
 PRINT USING "###.###"; MediumInfo1.Density
 COLOR WHITE, CYAN
 LOCATE 22, 19
 PRINT "Change to: "
 LOCATE 22, 30
 COLOR CYAN, CYAN
 PRINT TEMPBUF$
 COLOR WHITE, CYAN
 LOCATE 22, 30
 INPUT ; "", BUFFER$
 TEMPBUF$ = BUFFER$
 IF BUFFER$ <> "" THEN

 BUFFER! = VAL(BUFFER$)
 IF BUFFER! > 0 AND BUFFER! <= 999.999 THEN
 MediumInfo1.Density = BUFFER!
 VALID = 1
 ELSE
 LOCATE 23, 16
```

```
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 2, COLM - 3
COLOR YELLOW, BLUE
PRINT USING "###.###"; MediumInfo1.Density
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "4" THEN

CALL DISPINFOBOX
VALID = 0
TEMPBUF$ = ""
DO
COLOR YELLOW, CYAN
LOCATE 21, 20
PRINT "Viscosity: ";
PRINT USING "###.###"; MediumInfo1.Viscosity
COLOR WHITE, CYAN
LOCATE 22, 20
PRINT "Change to: "
LOCATE 22, 31
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 31
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$

IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 999.999 THEN
MediumInfo1.Viscosity = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 3, COLM - 1
COLOR YELLOW, BLUE
PRINT USING "###.###"; MediumInfo1.Viscosity
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "3" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 21
PRINT "Density:";
PRINT USING "###.###"; MediumInfo1.Density
COLOR WHITE, CYAN
LOCATE 22, 19
PRINT "Change to: "
LOCATE 22, 30
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 30
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
```

```
IF BUFFER! > 0 AND BUFFER! <= 999.999 THEN
MediumInfo1.Density = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 6, COLM - 3
COLOR YELLOW, BLUE
PRINT USING "###.###"; MediumInfo1.Density
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "5" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 16
PRINT "Hamaker Constant:";
PRINT USING "###.###^^^^"; MediumInfo1.Hamaker
COLOR WHITE, CYAN
LOCATE 22, 23
PRINT "Change to: "
LOCATE 22, 34
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 34
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN
BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 1E+30 THEN
MediumInfo1.Hamaker = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 1.0E+30"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 4, COLM + 6
COLOR YELLOW, BLUE
PRINT USING "###.###^^^^"; MediumInfo1.Hamaker
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "6" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 20
PRINT "Permeability:";
PRINT USING "####.###"; MediumInfo1.Perm
COLOR WHITE, CYAN
LOCATE 22, 23
PRINT "Change to: "
LOCATE 22, 35
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 35
INPUT ; "", BUFFER$
```

```
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 9999.999 THEN
MediumInfo1.Perm = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 9999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 5, COLM + 2
COLOR YELLOW, BLUE
PRINT USING "####.###"; MediumInfo1.Perm
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "7" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 16
PRINT "Temperature:";
PRINT USING "#####.###"; MediumInfo1.Temp
COLOR WHITE, CYAN
LOCATE 22, 18
PRINT "Change to: "
LOCATE 22, 29
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 29
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > -273.15 AND BUFFER! <= 9999.999 THEN
MediumInfo1.Temp = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between -273.150 and 9999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 6, COLM
COLOR YELLOW, BLUE
PRINT USING "#####.###"; MediumInfo1.Temp
CALL CLEARINFOBOX


ELSEIF CHOICE$ = CHR$(27) THEN 'Escape
OPT% = 0
DONE = 1
ELSEIF UCASE$(CHOICE$) = "H" THEN
CALL HELPMEDIUM
SCREEN 0, , 0
END IF
LOOP UNTIL DONE = 1

IF QUIT = 1 THEN
CLS
END
END IF
```

```
END SUB

DEFSNG A-Z
SUB GETPARTA (OPT%)

'-----------------------------------------------------

'=====================================================
' Program Name: GETPARTA
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'

'$INCLUDE: 'MENUPAR.INC'

OPT% = 0
DONE = 0
QUIT = 0

ROW = 7
COLM = 40

COLOR YELLOW, BLUE

LOCATE ROW, COLM + 2
PRINT PartInfoA.ComName

LOCATE ROW + 1, COLM + 4
PRINT PartInfoA.ChemName

LOCATE ROW + 2, COLM - 4
PRINT PartInfoA.Shape

LOCATE ROW + 3, COLM + 14
PRINT USING "##.###"; PartInfoA.MaxVel

LOCATE ROW + 4, COLM + 3
PRINT USING "##.###"; PartInfoA.MeanDiameter

LOCATE ROW + 5, COLM + 12
PRINT USING "##.###"; PartInfoA.SDevDiameter

LOCATE ROW + 6, COLM - 3
PRINT USING "###.###"; PartInfoA.Density

LOCATE ROW + 7, COLM + 9
PRINT USING "###"; PartInfoA.Number

LOCATE ROW + 8, COLM + 6
PRINT USING "###.###^^^^"; PartInfoA.Hamaker

BLANK$ = " "
DO
 CHOICE$ = INKEY$

 IF CHOICE$ = "1" THEN
 CALL DISPINFOBOX
 BUFFER$ = ""
 LOCATE 21, 18
 COLOR YELLOW, CYAN
 PRINT "Common Name: "; PartInfoA.ComName
 LOCATE 22, 20
 COLOR WHITE, CYAN
 PRINT "Change to: "
 LOCATE 22, 31
 COLOR YELLOW, CYAN
 INPUT ; "", BUFFER$
 IF BUFFER$ <> "" THEN PartInfoA.ComName = BUFFER$
```

```
LOCATE ROW, COLM + 2
COLOR YELLOW, BLUE
PRINT PartInfoA.ComName
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "2" THEN

CALL DISPINFOBOX
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 18
PRINT "Chemical Name: "; PartInfoA.ChemName
COLOR WHITE, CYAN
LOCATE 22, 22
PRINT "Change to: "
LOCATE 22, 33
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN PartInfoA.ChemName = BUFFER$
LOCATE ROW + 1, COLM + 4
COLOR YELLOW, BLUE
PRINT PartInfoA.ChemName
CALL CLEARINFOBOX


ELSEIF CHOICE$ = "3" THEN

CALL DISPINFOBOX
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 24
PRINT "Shape: "; PartInfoA.Shape
COLOR WHITE, CYAN
LOCATE 22, 20
PRINT "Change to: "
LOCATE 22, 31
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN PartInfoA.Shape = BUFFER$
LOCATE ROW + 2, COLM - 4
COLOR YELLOW, BLUE
PRINT PartInfoA.Shape
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "4" THEN
 CALL DISPINFOBOX
 VALID = 0
 TEMPBUF$ = ""
 DO
 COLOR YELLOW, CYAN
 LOCATE 21, 18
 PRINT "Average Initial Velocity: ";
 PRINT USING "##.###"; PartInfoA.MaxVel
 COLOR WHITE, CYAN
 LOCATE 22, 25
 PRINT "Change to: "
 LOCATE 22, 36
 COLOR CYAN, CYAN
 PRINT TEMPBUF$
 COLOR WHITE, CYAN
 LOCATE 22, 36
 INPUT ; "", BUFFER$
 TEMPBUF$ = BUFFER$

 IF BUFFER$ <> "" THEN

 BUFFER = VAL(BUFFER$)
 IF BUFFER > 0 AND BUFFER <= 50! THEN
 PartInfoA.MaxVel = BUFFER
 VALID = 1
 ELSE
 LOCATE 23, 15
 COLOR YELLOW, CYAN
 PRINT "Valid input is a real number between 0 and 50.0"
 END IF
 ELSE
 VALID = 1
 END IF
 LOOP UNTIL VALID = 1
```

```
LOCATE ROW + 3, COLM + 14
COLOR YELLOW, BLUE
PRINT USING "##.###"; PartInfoA.MaxVel
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "5" THEN
 CALL DISPINFOBOX
 VALID = 0
 TEMPBUF$ = ""
 DO
 COLOR YELLOW, CYAN
 LOCATE 21, 19
 PRINT "Mean Diameter:";
 PRINT USING "##.###"; PartInfoA.MeanDiameter
 COLOR WHITE, CYAN
 LOCATE 22, 23
 PRINT "Change to: "
 LOCATE 22, 34
 COLOR CYAN, CYAN
 PRINT TEMPBUF$
 COLOR WHITE, CYAN
 LOCATE 22, 34
 INPUT ; "", BUFFER$
 TEMPBUF$ = BUFFER$

 IF BUFFER$ <> "" THEN

 BUFFER = VAL(BUFFER$)
 IF BUFFER > 0 AND BUFFER <= 2! THEN
 PartInfoA.MeanDiameter = BUFFER
 VALID = 1
 ELSE
 LOCATE 23, 15
 COLOR YELLOW, CYAN
 PRINT "Valid input is a real number between 0 and 2.0"
 END IF
 ELSE
 VALID = 1
 END IF

LOOP UNTIL VALID = 1

LOCATE ROW + 4, COLM + 3
COLOR YELLOW, BLUE
PRINT USING "##.###"; PartInfoA.MeanDiameter
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "6" THEN

 CALL DISPINFOBOX
 VALID = 0
 TEMPBUF$ = ""
 DO
 COLOR YELLOW, CYAN
 LOCATE 21, 13
 PRINT "Diameter Standard Dev.:";
 PRINT USING "##.###"; PartInfoA.SDevDiameter
 COLOR WHITE, CYAN
 LOCATE 22, 26
 PRINT "Change to: "
 LOCATE 22, 37
 COLOR CYAN, CYAN
 PRINT TEMPBUF$
 COLOR WHITE, CYAN
 LOCATE 22, 37
 INPUT ; "", BUFFER$
 TEMPBUF$ = BUFFER$

 IF BUFFER$ <> "" THEN

 BUFFER = VAL(BUFFER$)
 IF BUFFER > 0 AND BUFFER <= 1! THEN
 PartInfoA.SDevDiameter = BUFFER
 VALID = 1
 ELSE
 LOCATE 23, 15
 COLOR YELLOW, CYAN
 PRINT "Valid input is a real number between 0 and 1.0"
 END IF
```

```
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 5, COLM + 12
COLOR YELLOW, BLUE
PRINT USING "##.###"; PartInfoA.SDevDiameter
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "7" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 21
PRINT "Density:";
PRINT USING "###.###"; PartInfoA.Density
COLOR WHITE, CYAN
LOCATE 22, 19
PRINT "Change to: "
LOCATE 22, 30
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 30
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER = VAL(BUFFER$)
IF BUFFER > 0 AND BUFFER <= 999.999 THEN
PartInfoA.Density = BUFFER
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 6, COLM - 3
COLOR YELLOW, BLUE
PRINT USING "###.###"; PartInfoA.Density
CALL CLEARINFOBOX


ELSEIF CHOICE$ = "8" THEN

VALID = 0
TEMPBUF$ = ""
CALL DISPINFOBOX

IF Simulation = 1 THEN
NUMPART = 60
IF PartInfoA.Number > NUMPART THEN PartInfoA.Number = NUMPART
ELSEIF Simulation = 2 THEN
NUMPART = 120
END IF

DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 16
PRINT "Number of Particles:";
PRINT USING "###"; PartInfoA.Number
LOCATE 22, 26
COLOR WHITE, CYAN
PRINT "Change to: "
COLOR CYAN, CYAN
LOCATE 22, 37
PRINT TEMPBUF$
COLOR WHITE, CYAN
```

```
LOCATE 22, 37
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER = VAL(BUFFER$)

IF Simulation = 1 THEN
NUMPART = 60
IF PartInfoA.Number > NUMPART THEN PartInfoA.Number = NUMPART
ELSEIF Simulation = 2 THEN
NUMPART = 120
END IF

IF BUFFER > 0 AND BUFFER <= NUMPART THEN
PartInfoA.Number = BUFFER
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN

IF Simulation = 1 THEN
PRINT "Valid input is an integer number between 1 and 60"
ELSEIF Simulation = 2 THEN
PRINT "Valid input is an integer number between 1 and 120"
END IF

END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 7, COLM + 9
COLOR YELLOW, BLUE
PRINT USING "###"; PartInfoA.Number
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "9" THEN

VALID = 0
TEMPBUF$ = ""
CALL DISPINFOBOX

DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 16
PRINT "Hamaker Constant:";
PRINT USING "###.###^^^^"; PartInfoA.Hamaker
LOCATE 22, 23
COLOR WHITE, CYAN
PRINT "Change to: "
COLOR CYAN, CYAN
LOCATE 22, 37
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 34
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER = VAL(BUFFER$)
IF BUFFER > 0 AND BUFFER <= 1E+30 THEN
PartInfoA.Hamaker = BUFFER
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is an integer number between 1 and 1.0E+30"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 8, COLM + 6
COLOR YELLOW, BLUE
```

```
PRINT USING "###.###^^^^"; PartInfoA.Hamaker
CALL CLEARINFOBOX
ELSEIF UCASE$(CHOICE$) = "H" THEN
CALL HELPPARTA
SCREEN 0, , 0
ELSEIF CHOICE$ = CHR$(27) THEN 'Escape
OPT% = 0
DONE = 1
END IF
LOOP UNTIL DONE = 1

IF QUIT = 1 THEN
 CLS
 END
END IF


END SUB

DEFINT A-Z
SUB GETPARTB (OPT%)

'------------------------------------------------------------

'============================================================
' Program Name: GETPARTB
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'============================================================
'
```

```
'$INCLUDE: 'MENUPAR.INC'

OPT% = 0
DONE = 0
QUIT = 0

ROW = 7
COLM = 40

COLOR YELLOW, BLUE

LOCATE ROW, COLM + 2
PRINT PartInfoB.ComName

LOCATE ROW + 1, COLM + 4
PRINT PartInfoB.ChemName

LOCATE ROW + 2, COLM - 4
PRINT PartInfoB.Shape

LOCATE ROW + 3, COLM + 14
PRINT USING "##.###"; PartInfoB.MaxVel

LOCATE ROW + 4, COLM + 3
PRINT USING "##.###"; PartInfoB.MeanDiameter

LOCATE ROW + 5, COLM + 12
PRINT USING "##.###"; PartInfoB.SDevDiameter

LOCATE ROW + 6, COLM - 3
PRINT USING "###.###"; PartInfoB.Density

LOCATE ROW + 7, COLM + 9
PRINT USING "###"; PartInfoB.Number

LOCATE ROW + 8, COLM + 6
PRINT USING "###.###^^^^"; PartInfoB.Hamaker

BLANK$ = " "
```

```
DO
CHOICE$ = INKEY$

IF CHOICE$ = "1" THEN
CALL DISPINFOBOX
BUFFER$ = ""
LOCATE 21, 18
COLOR YELLOW, CYAN
PRINT "Common Name: "; PartInfoB.ComName
LOCATE 22, 20
COLOR WHITE, CYAN
PRINT "Change to: "
LOCATE 22, 31
COLOR YELLOW, CYAN
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN PartInfoB.ComName = BUFFER$
LOCATE ROW, COLM + 2
COLOR YELLOW, BLUE
PRINT PartInfoB.ComName
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "2" THEN

CALL DISPINFOBOX
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 18
PRINT "Chemical Name: "; PartInfoB.ChemName
COLOR WHITE, CYAN
LOCATE 22, 22
PRINT "Change to: "
LOCATE 22, 33
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN PartInfoB.ChemName = BUFFER$
LOCATE ROW + 1, COLM + 4
COLOR YELLOW, BLUE
PRINT PartInfoB.ChemName
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "3" THEN

CALL DISPINFOBOX
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 24
PRINT "Shape: "; PartInfoB.Shape
COLOR WHITE, CYAN
LOCATE 22, 20
PRINT "Change to: "
LOCATE 22, 31
INPUT ; "", BUFFER$
IF BUFFER$ <> "" THEN PartInfoB.Shape = BUFFER$
LOCATE ROW + 2, COLM - 4
COLOR YELLOW, BLUE
PRINT PartInfoB.Shape
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "4" THEN
CALL DISPINFOBOX
VALID = 0
TEMPBUF$ = ""
DO
COLOR YELLOW, CYAN
LOCATE 21, 18
PRINT "Average Initial Velocity: ";
PRINT USING "##.###"; PartInfoB.MaxVel
COLOR WHITE, CYAN
LOCATE 22, 25
PRINT "Change to: "
LOCATE 22, 36
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 36
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
```

```
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 50! THEN
PartInfoB.MaxVel = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 50.0"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 3, COLM + 14
COLOR YELLOW, BLUE
PRINT USING "##.###"; PartInfoB.MaxVel
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "5" THEN
CALL DISPINFOBOX
VALID = 0
TEMPBUF$ = ""
DO
COLOR YELLOW, CYAN
LOCATE 21, 19
PRINT "Mean Diameter:";
PRINT USING "##.###"; PartInfoB.MeanDiameter
COLOR WHITE, CYAN
LOCATE 22, 23
PRINT "Change to: "
LOCATE 22, 34
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 34
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$

IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 2! THEN
PartInfoB.MeanDiameter = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 2.0"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 4, COLM + 3
COLOR YELLOW, BLUE
PRINT USING "##.###"; PartInfoB.MeanDiameter
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "6" THEN

CALL DISPINFOBOX
VALID = 0
TEMPBUF$ = ""
DO
COLOR YELLOW, CYAN
LOCATE 21, 13
PRINT "Diameter Standard Dev.:";
PRINT USING "##.###"; PartInfoB.SDevDiameter
COLOR WHITE, CYAN
LOCATE 22, 26
PRINT "Change to: "
LOCATE 22, 37
COLOR CYAN, CYAN
PRINT TEMPBUF$
```

```
COLOR WHITE, CYAN
LOCATE 22, 37
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$

IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 1! THEN
PartInfoB.SDevDiameter = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 1.0"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 5, COLM + 12
COLOR YELLOW, BLUE
PRINT USING "##.###"; PartInfoB.SDevDiameter
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "7" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 21
PRINT "Density:";
PRINT USING "###.###"; PartInfoB.Density
COLOR WHITE, CYAN
LOCATE 22, 19
PRINT "Change to: "
LOCATE 22, 30
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 30
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 999.999 THEN
PartInfoB.Density = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 999.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 6, COLM - 3
COLOR YELLOW, BLUE
PRINT USING "###.###"; PartInfoB.Density
CALL CLEARINFOBOX


ELSEIF CHOICE$ = "8" THEN

VALID = 0
TEMPBUF$ = ""
CALL DISPINFOBOX

DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 16
```

```
PRINT "Number of Particles:";
PRINT USING "###"; PartInfoB.Number
LOCATE 22, 26
COLOR WHITE, CYAN
PRINT "Change to: "
COLOR CYAN, CYAN
LOCATE 22, 37
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 37
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 60 THEN
PartInfoB.Number = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is an integer number between 1 and 60"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 7, COLM + 9
COLOR YELLOW, BLUE
PRINT USING "###"; PartInfoB.Number
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "9" THEN

VALID = 0
TEMPBUF$ = ""
CALL DISPINFOBOX

DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 16
PRINT "Hamaker Constant:";
PRINT USING "###.###^^^^"; PartInfoB.Hamaker
LOCATE 22, 23
COLOR WHITE, CYAN
PRINT "Change to: "
COLOR CYAN, CYAN
LOCATE 22, 37
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 34
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 1E+30 THEN
PartInfoB.Hamaker = BUFFER!
VALID = 1
ELSE
LOCATE 23, 15
COLOR YELLOW, CYAN
PRINT "Valid input is an integer number between 1 and 1.0E+30"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 8, COLM + 6
COLOR YELLOW, BLUE
PRINT USING "###.###^^^^"; PartInfoB.Hamaker
CALL CLEARINFOBOX

ELSEIF UCASE$(CHOICE$) = "H" THEN
CALL HELPPARTB
```

SCREEN 0, 0

```
 SCREEN 0, , 0

 ELSEIF CHOICE$ = CHR$(27) THEN 'Escape
 OPT% = 0
 DONE = 1
 END IF
LOOP UNTIL DONE = 1

IF QUIT = 1 THEN
 CLS
 END
END IF

END SUB

DEFSNG A-Z
SUB GETSIM1 (OPT%)

'-------------------------------------------------------

'=======================================================
' Program Name: GETSIM1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

OPT% = 0
DONE = 0
QUIT = 0

DO
 CHOICE$ = INKEY$

 IF CHOICE$ = "1" THEN
 OPT% = 1
 DONE = 1
 ELSEIF CHOICE$ = "2" THEN
 OPT% = 2
 DONE = 1
 ELSEIF CHOICE$ = "3" THEN
 OPT% = 3
 DONE = 1
 ELSEIF UCASE$(CHOICE$) = "R" THEN
 OPT% = 4
 DONE = 1
 ELSEIF UCASE$(CHOICE$) = "H" THEN
 CALL HELPSIM
 SCREEN 0, , 0
 ELSEIF CHOICE$ = CHR$(27) THEN
 OPT% = 0
 DONE = 1
 END IF
LOOP WHILE DONE = 0

IF QUIT = 1 THEN
 CLS
 END
END IF

END SUB

DEFINT A-Z
SUB GETSURFACE1 (OPT%)

'-------------------------------------------------------

'=======================================================
' Program Name: GETSURFACE1
'
```

* Description: This subroutine retrieves info

```
' Description: This subroutine retrieves information for the
' particle information menu.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

'$INCLUDE: 'MENUPAR.INC'

OPT% = 0
DONE = 0
QUIT = 0

ROW = 6
COLM = 40

COLOR YELLOW, CYAN

LOCATE ROW, COLM
PRINT USING "##.##"; pHInfo1

LOCATE ROW + 1, COLM + 13
PRINT USING "###.###"; ZetaInfoA

IF Simulation = 1 THEN
 LOCATE ROW + 2, COLM + 13
 PRINT USING "###.###"; ZetaInfoB
ELSEIF Simulation = 2 THEN
 LOCATE ROW + 2, COLM + 13
 PRINT USING "###.###"; ZetaInfoF
END IF

DO

CHOICE$ = INKEY$

IF CHOICE$ = "1" THEN
VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 26
PRINT "pH: ";
PRINT USING "##.##"; pHInfo1
COLOR WHITE, CYAN
LOCATE 22, 19
PRINT "Change to:"
LOCATE 22, 30
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 30
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > 0 AND BUFFER! <= 14! THEN
pHInfo1 = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between 0 and 14.0"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW, COLM
```

PRINTED IN THE UNITED STATES OF AMERICA

```
COLOR YELLOW, CYAN
PRINT USING "##.##"; pHInfo1
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "2" THEN

VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 10
PRINT "Zeta Potential of Particle A: ";
PRINT USING "###.###"; ZetaInfoA
COLOR WHITE, CYAN
LOCATE 22, 29
PRINT "Change to: "
LOCATE 22, 40
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 40
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
IF BUFFER! > -99.999 AND BUFFER! <= 99.999 THEN
ZetaInfoA = BUFFER!
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between -99.999 and 99.999"
END IF
ELSE
VALID = 1
END IF

LOOP UNTIL VALID = 1

LOCATE ROW + 1, COLM + 13
COLOR YELLOW, CYAN
PRINT USING "###.###"; ZetaInfoA
CALL CLEARINFOBOX

ELSEIF CHOICE$ = "3" THEN

VALID = 0
CALL DISPINFOBOX
TEMPBUF$ = ""
DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 21, 10

IF Simulation = 1 THEN

PRINT "Zeta Potential of Particle B: ";
PRINT USING "###.###"; ZetaInfoB
ELSEIF Simulation = 2 THEN
PRINT "Zeta Potential of the Fiber: ";
PRINT USING "###.###"; ZetaInfoF
END IF

COLOR WHITE, CYAN
LOCATE 22, 29
PRINT "Change to: "
LOCATE 22, 40
COLOR CYAN, CYAN
PRINT TEMPBUF$
COLOR WHITE, CYAN
LOCATE 22, 40
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)
```

```
IF BUFFER! > -99.999 AND BUFFER! <= 99.999 THEN
IF Simulation = 1 THEN
ZetaInfoB = BUFFER!
ELSEIF Simulation = 2 THEN
ZetaInfoF = BUFFER!
END IF
VALID = 1
ELSE
LOCATE 23, 16
COLOR YELLOW, CYAN
PRINT "Valid input is a real number between -99.999 and 99.999"
END IF
ELSE
VALID = 1
END IF
LOOP UNTIL VALID = 1

LOCATE ROW + 2, COLM + 13
COLOR YELLOW, CYAN
IF Simulation = 1 THEN
PRINT USING "###.###"; ZetaInfoB
ELSEIF Simulation = 2 THEN
PRINT USING "###.###"; ZetaInfoF
END IF

CALL CLEARINFOBOX

ELSEIF CHOICE$ = CHR$(27) THEN 'Escape
OPT% = 0
DONE = 1
ELSEIF UCASE$(CHOICE$) = "H" THEN
CALL HELPSURFACE
SCREEN 0, , 0
END IF
LOOP UNTIL DONE = 1

IF QUIT = 1 THEN
 CLS
 END
END IF


END SUB

SUB HELPDYN

'----------------------------------------------------

'====================================================
' Program Name: HELPDYN
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'====================================================
'

HELPMES$(1) = " Dynamic Model Menu"
HELPMES$(2) = " "
HELPMES$(3) = " Press the number next to the model that is desired."
HELPMES$(4) = " "
HELPMES$(5) = "The 'DLVO theory' model "
HELPMES$(6) = " (acronym: Derjaguin - Landau - Verwey - Overbeek) "
HELPMES$(7) = "is the theory stating that the stability of a colloidal suspension "
HELPMES$(8) = "is based on the sum of the electrostatic repulsions due to the over-
lap"
HELPMES$(9) = "of electrical double layers plus the attractive potential due to the "
HELPMES$(10) = "London - van der Waals forces. "
HELPMES$(11) = " "
HELPMES$(12) = "The Acid/Base model "
HELPMES$(13) = " "
HELPMES$(14) = " "
```

HELPMES$(15) = "The Random model "

```
HELPMES$(15) = "The Random model "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = "The time increment specifies the interval between calculations."
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPELECTRO

'-------------------------------------------------------

'=======================================================
' Program Name: HELPELECTRO
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

HELPMES$(1) = " Electrolyte Information Menu"
HELPMES$(2) = " "
HELPMES$(3) = " "
HELPMES$(4) = " Press the number next to the option to be changed. "
HELPMES$(5) = " "
HELPMES$(6) = " "
HELPMES$(7) = " "
HELPMES$(8) = " "
HELPMES$(9) = " "
HELPMES$(10) = " "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPFIBER

'-------------------------------------------------------

'=======================================================
' Program Name: HELPFIBER
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

HELPMES$(1) = " FIber Information Menu"
HELPMES$(2) = " "
HELPMES$(3) = " "
HELPMES$(4) = " Press the number next to the option to be changed. "
HELPMES$(5) = " "
HELPMES$(6) = " "
```

HELPMES$(7) = "...

```
HELPMES$(7) = " "
HELPMES$(8) = " "
HELPMES$(9) = " "
HELPMES$(10) = " "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPMAIN

'-------------------------------------------------------

'=======================================================
' Program Name: HELPMAIN
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

HELPMES$(1) = " Welcome to the Colloidal Suspension Simulator. "
HELPMES$(2) = " "
HELPMES$(3) = " This program simulates the dynamics of colloidal particles "
HELPMES$(4) = "while they are in suspension. The main menu allows you to choose"
HELPMES$(5) = "a simulation type and to save or load a configuration. The first "
HELPMES$(6) = "simulation involves the interactions between two types of particles,"
HELPMES$(7) = "i.e., between particles of type A and particles of type B. The "
HELPMES$(8) = "second simulation involves the interactions between one type of "
HELPMES$(9) = "particle and a fiber. "
HELPMES$(10) = " "
HELPMES$(11) = "Press the number next to the simulation type that is desired. "
HELPMES$(12) = " "
HELPMES$(13) = " A configuration is the set of current values that describe "
HELPMES$(14) = "information about the particles, the suspension medium, the fiber,"
HELPMES$(15) = "the electrolyte, the surface parameters and the choice of a "
HELPMES$(16) = "dynamic model. When the program is run for the first time, a
default"
HELPMES$(17) = "configuration is loaded. A configuration may be saved or loaded "
HELPMES$(18) = "by pressing the number next to the 'Save' or 'Load' options. "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPMATRL

'-------------------------------------------------------

'=======================================================
' Program Name: HELPMATRL
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
```

```
'=============================================================
'

HELPMES$(1) = " Material Parameters Menu"
HELPMES$(2) = " "
HELPMES$(3) = " "
HELPMES$(4) = " "
HELPMES$(5) = " Press the number of the option desired. "
HELPMES$(6) = " "
HELPMES$(7) = " "
HELPMES$(8) = " "
HELPMES$(9) = " "
HELPMES$(10) = " "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPMEDIUM

'-------------------------------------------------------------

'=============================================================
' Program Name: HELPMEDIUM
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

HELPMES$(1) = " Medium Information Menu"
HELPMES$(2) = " "
HELPMES$(3) = " "
HELPMES$(4) = " Press the number next to the option to be changed. "
HELPMES$(5) = " "
HELPMES$(6) = " "
HELPMES$(7) = " "
HELPMES$(8) = " "
HELPMES$(9) = " "
HELPMES$(10) = " "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPPARTA

'-------------------------------------------------------------

'=============================================================
' Program Name: HELPPARTA
```

Description

```
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'================================================
'


HELPMES$(1) = " Particle A Information Menu"
HELPMES$(2) = " "
HELPMES$(3) = " "
HELPMES$(4) = " Press the number next to the option to be changed. "
HELPMES$(5) = " "
HELPMES$(6) = " "
HELPMES$(7) = " "
HELPMES$(8) = " "
HELPMES$(9) = " "
HELPMES$(10) = " "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPPARTB
```

```
'-------------------------------------------------

'================================================
' Program Name: HELPPARTB
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'================================================
'


HELPMES$(1) = " Particle B Information Menu"
HELPMES$(2) = " "
HELPMES$(3) = " "
HELPMES$(4) = " Press the number next to the option to be changed. "
HELPMES$(5) = " "
HELPMES$(6) = " "
HELPMES$(7) = " "
HELPMES$(8) = " "
HELPMES$(9) = " "
HELPMES$(10) = " "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "
```

```
CALL DISPHELP(HELPMES$())
```

```
CALL DISPHELP(HELPMES$())

END SUB

SUB HELPSIM

'-------------------------------------------------

'=================================================
' Program Name: HELPSIM
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=================================================
'

IF Simulation = 1 THEN
 HELPMES$(1) = " Particle A - Particle B Menu "
ELSEIF Simulation = 2 THEN
 HELPMES$(1) = " Particle A - Fiber Menu "
END IF
HELPMES$(2) = " "
HELPMES$(3) = " Press the number next to the desired option. "
HELPMES$(4) = " "
HELPMES$(5) = " Press 1 to modify the Material parameters. "
HELPMES$(6) = " Press 2 to modify the surface parameters. "
HELPMES$(7) = " Press 3 to choose a dynamic model or to change the time "
HELPMES$(8) = " increment for which the model will run. "
HELPMES$(9) = " Press R to run the cssrun program that will compute "
HELPMES$(10) = " the new positions for the system. "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB HELPSURFACE

'-------------------------------------------------

'=================================================
' Program Name: HELPSURFACE
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=================================================
'

HELPMES$(1) = " Surface Information Menu"
HELPMES$(2) = " "
HELPMES$(3) = " "
HELPMES$(4) = " Press the number next to the option to be changed. "
HELPMES$(5) = " "
HELPMES$(6) = " "
HELPMES$(7) = " "
HELPMES$(8) = " "
```

HELPMES$(9) = " "
HELPMES$(10) = " "

```
HELPMES$(9) = " "
HELPMES$(10) = " "
HELPMES$(11) = " "
HELPMES$(12) = " "
HELPMES$(13) = " "
HELPMES$(14) = " "
HELPMES$(15) = " "
HELPMES$(16) = " "
HELPMES$(17) = " "
HELPMES$(18) = " "
HELPMES$(19) = " "

CALL DISPHELP(HELPMES$())

END SUB

SUB LOADATA

'-------------------------------------------------------

'=======================================================
' Program Name: LOADDATA
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

CALL GETFILENAM(FILENAM$)

IF LEN(FILENAM$) > 8 THEN
 FOR LL = 1 TO 19
 MESSAGE$(LL) = " "
 NEXT LL
 MESSAGE$(6) = " Invalid File Name"
 MESSAGE$(8) = " File names must be eight characters or less."

 CALL DISPERR(MESSAGE$())
 FILENAM$ = ""
END IF

IF FILENAM$ <> "" THEN

FILENAM$ = FILENAM$ + ".CSS"
FILEN$ = FILENAM$ + CHR$(0)
CALL EXIST(FILEN$, FILEXISTS%)

IF FILEXISTS% THEN
OPEN FILENAM$ FOR RANDOM ACCESS READ WRITE AS #2

GET #2, 1, PartInfoA
GET #2, 2, PartInfoB
GET #2, 3, MediumInfo1
GET #2, 4, ElectroInfo1
GET #2, 5, FiberInfo
GET #2, 6, MENULVL
GET #2, 7, pHInfo1
GET #2, 8, ZetaInfoA
GET #2, 9, ZetaInfoB
GET #2, 10, ZetaInfoF
GET #2, 11, DYNMDL
GET #2, 12, Simulation
GET #2, 13, INITDAT

CLOSE #2

ELSE
FOR LL = 1 TO 19
MESSAGE$(LL) = " "
NEXT LL
```

```
MESSAGE$(6) = " The file: " + FILENAM$
MESSAGE$(7) = "
```

```
MESSAGE$(6) = " The file: " + FILENAM$
MESSAGE$(7) = " Does not exist"
CALL DISPERR(MESSAGE$())
END IF
END IF

END SUB

DEFSNG A-Z
SUB MATERIAL1

'-----------------------------------------------------------

'=============================================================
' Program Name: MATERIAL1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

DONE = 0

DO
'================================
' Display the Material Parameters Menu for Model One.
'================================

IF MENULVL = 2 THEN
CALL DISPMATERIAL1
ELSE
CALL DISPMAIN
CALL DISPSIM1
CALL DISPMATERIAL1
MENULVL = 3
END IF

'================================
' Accept and verify the input to the Material Parameters Menu.
'================================

CALL GETMATERIAL1(OPT%)

MENULVL = 3

IF OPT% = 1 THEN
MENULVL = 3
CALL PARTA
ELSEIF OPT% = 2 THEN
IF Simulation = 1 THEN
CALL PARTB
ELSEIF Simulation = 2 THEN
CALL FIBER
END IF
ELSEIF OPT% = 3 THEN
CALL MEDIUM1
ELSEIF OPT% = 4 THEN
CALL ELECTRO1
ELSEIF OPT% = 0 THEN ' Return to the previous menu
DONE = 1
END IF

LOOP UNTIL DONE = 1

END SUB

DEFINT A-Z
SUB MEDIUM1

'-----------------------------------------------------------
```

Program Name: MUSH

```
'=============================================================
' Program Name: MEDIUM1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

DONE = 0

DO
'==================================
' Display the Particle Information Menu for Model One.
'==================================

 IF MENULVL = 3 THEN
 CALL DISPMEDIUM1
 ELSE
 CALL DISPMAIN
 CALL DISPSIM1
 CALL DISPMATERIAL1
 CALL DISPMEDIUM1

 END IF

'==================================
' Accept and verify the input to the Particle Information Menu.
'==================================

 CALL GETMEDIUM1(OPT%)

 IF OPT% = 0 THEN ' Return to the previous menu
 DONE = 1
 END IF

LOOP UNTIL DONE = 1

END SUB

DEFSNG A-Z
SUB PARTA

'-------------------------------------------------------

'=============================================================
' Program Name: PARTA
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

DONE = 0

DO
'==================================
' Display the Particle Information Menu for Model One.
'==================================

 IF MENULVL = 3 THEN
 CALL DISPPARTA
 ELSE
 CALL DISPMAIN
 CALL DISPSIM1
```

```
CALL DISPMATERIAL1
CALL DISPPARTA

END IF

'=====================================
' Accept and verify the input to the Particle Information Menu.
'=====================================

CALL GETPARTA(OPT%)

IF OPT% = 0 THEN ' Return to the previous menu
DONE = 1
END IF

LOOP UNTIL DONE = 1

END SUB

DEFINT A-Z
SUB PARTB

'-----------------------------------------------------

'=====================================================
' Program Name: PARTB
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'

DONE = 0

DO
'=====================================
' Display the Particle Information Menu for Model One.
'=====================================

IF MENULVL = 3 THEN
CALL DISPPARTB
ELSE
CALL DISPMAIN
CALL DISPSIM1
CALL DISPMATERIAL1
CALL DISPPARTB

END IF

'=====================================
' Accept and verify the input to the Particle Information Menu.
'=====================================

CALL GETPARTB(OPT%)

IF OPT% = 0 THEN ' Return to the previous menu
DONE = 1
END IF

LOOP UNTIL DONE = 1

END SUB

SUB READSTAT

'-----------------------------------------------------

'=============================================
' Program Name: READSTAT
'
' Description:
```

```
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=========================================================
'

TFIL$ = "CSSTEMP"
TFILE$ = "CSSTEMP" + CHR$(0)

CALL EXIST(TFILE$, FILEXISTS%)

IF FILEXISTS% THEN
 OPEN TFIL$ FOR RANDOM ACCESS READ WRITE AS #1
 GET #1, 13, INITDAT
 CLOSE #1
END IF

END SUB

SUB RUNDISPLAY

DISPFIL$ = CSSDISP$ + CHR$(0)
CALL EXIST(DISPFIL$, FILEXISTS%)

IF FILEXISTS% THEN
 RUN CSSDISP$
ELSE
 FOR LL = 1 TO 19
 MESSAGE$(LL) = " "
 NEXT LL
 MESSAGE$(2) = " File Exist Error "
 MESSAGE$(4) = " The file CSSDISP.EXE does not exist"
 CALL DISPERR(MESSAGE$())
 SCREEN 0, , 0
END IF

END SUB

SUB SAVEDATA

'---------------------------------------------------

'=========================================================
' Program Name: SAVEDATA
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=========================================================
'

CALL GETFILENAM(FILENAM$)

IF LEN(FILENAM$) > 8 THEN
 FOR LL = 1 TO 19
 MESSAGE$(LL) = " "
 NEXT LL
 MESSAGE$(6) = " Invalid File Name"
 MESSAGE$(8) = " File names must be eight characters or less."
 CALL DISPERR(MESSAGE$())
 FILENAM$ = ""
END IF

IF FILENAM$ <> "" THEN

 FILENAM$ = FILENAM$ + ".CSS"
```

OPEN FILENAME FOR

```
OPEN FILENAM$ FOR RANDOM ACCESS READ WRITE AS #2

PUT #2, 1, PartInfoA
PUT #2, 2, PartInfoB
PUT #2, 3, MediumInfo1
PUT #2, 4, ElectroInfo1
PUT #2, 5, FiberInfo
PUT #2, 6, MENULVL
PUT #2, 7, pHInfo1
PUT #2, 8, ZetaInfoA
PUT #2, 9, ZetaInfoB
PUT #2, 10, ZetaInfoF
PUT #2, 11, DYNMDL
PUT #2, 12, Simulation
PUT #2, 13, INITDAT

CLOSE #2

END IF

END SUB

DEFSNG A-Z
SUB SIM1

'-----------------------------------------------------

'=====================================================
' Program Name: SIM1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'
DONE = 0

DO

'=================================
' Display the Model 1 Menu.
'=================================

IF MENULVL = 1 THEN
CALL DISPSIM1
ELSE
CALL DISPMAIN
CALL DISPSIM1
END IF

'=================================
' Accept and verify the input to the Model One menu
'=================================

CALL GETSIM1(OPT%)

MENULVL = 2

IF OPT% = 1 THEN
CALL MATERIAL1
ELSEIF OPT% = 2 THEN
CALL SURFACE1
ELSEIF OPT% = 3 THEN
CALL DYNMODEL
ELSEIF OPT% = 4 THEN
DYNFIL$ = CSSRUN$ + CHR$(0)
CALL EXIST(DYNFIL$, FILEXISTS%)

IF FILEXISTS% THEN
INITDAT = 1
CALL TRANSDATA
RUN CSSRUN$
```

```
ELSE
FOR LL = 1 TO 19
MESSAGE$(LL) = " "
NEXT LL
MESSAGE$(2) = " File Exist Error "
MESSAGE$(4) = " The file CSSRUN.EXE does not exist"
CALL DISPERR(MESSAGE$())
SCREEN 0, , 0

END IF

ELSEIF OPT% = 5 THEN

DISPFIL$ = CSSDISP$ + CHR$(0)
CALL EXIST(DISPFIL$, FILEXISTS%)

IF FILEXISTS% THEN
RUN CSSDISP$
ELSE
FOR LL = 1 TO 19
MESSAGE$(LL) = " "
NEXT LL
MESSAGE$(2) = " File Exist Error "
MESSAGE$(4) = " The file CSSDISP.EXE does not exist"
CALL DISPERR(MESSAGE$())
SCREEN 0, , 0

END IF

ELSEIF OPT% = 0 THEN

DONE = 1

END IF

LOOP UNTIL DONE = 1

END SUB
```

```
DEFINT A-Z
SUB SURFACE1

'-----------------------------------------------------

'=====================================================
' Program Name: SURFACE1
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'

DONE = 0

DO
'==================================
' Display the Particle Information Menu for Model One.
'==================================

IF MENULVL = 2 THEN
CALL DISPSURFACE1
ELSE
CALL DISPMAIN
CALL DISPSIM1
CALL DISPSURFACE1
END IF

'==================================
' Accept and verify the input to the Particle Information Menu.
'==================================
```

```
CALL GETSURFACE1(OPT%)

IF OPT% = 0 THEN ' Return to the previous menu
DONE = 1
END IF

LOOP UNTIL DONE = 1


END SUB

DEFSNG A-Z
SUB TRANSDATA

'-----------------------------------------------------

'=====================================================
' Program Name: TRANSDATA
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'

TFIL$ = "CSSTEMP"
OPEN TFIL$ FOR RANDOM ACCESS READ WRITE AS #1

 PUT #1, 1, PartInfoA
 PUT #1, 2, PartInfoB
 PUT #1, 3, MediumInfo1
 PUT #1, 4, ElectroInfo1
 PUT #1, 5, FiberInfo
 PUT #1, 6, MENULVL
 PUT #1, 7, pHInfo1
 PUT #1, 8, ZetaInfoA
 PUT #1, 9, ZetaInfoB
 PUT #1, 10, ZetaInfoF
 PUT #1, 11, DYNMDL
 PUT #1, 12, Simulation
 PUT #1, 13, INITDAT
 PUT #1, 14, TimeIncr

CLOSE #1

END SUB

DEFINT A-Z
SUB TRANSDATAG

'-----------------------------------------------------

'=====================================================
' Program Name: TRANSDATAG
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'

TFIL$ = "CSSTEMP"
OPEN TFIL$ FOR RANDOM ACCESS READ WRITE AS #1

 GET #1, 1, PartInfoA
 GET #1, 2, PartInfoB
```

```
 GET #1, 3, MediumInfo1
 GET #1, 4, ElectroInfo1
 GET #1, 5, FiberInfo
 GET #1, 6, MENULVL
 GET #1, 7, pHInfo1
 GET #1, 8, ZetaInfoA
 GET #1, 9, ZetaInfoB
 GET #1, 10, ZetaInfoF
 GET #1, 11, DYNMDL
 GET #1, 12, Simulation
 GET #1, 13, INITDAT
 GET #1, 14, TimeIncr

CLOSE #1

END SUB

SUB WRITESTAT

'-------------------------------------------------------

'=======================================================
' Program Name: WRITESTAT
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

TFIL$ = "CSSTEMP"
TFILE$ = "CSSTEMP" + CHR$(0)

INITDAT = 0

CALL EXIST(TFILE$, FILEXISTS%)

IF FILEXISTS% THEN
 OPEN TFIL$ FOR RANDOM ACCESS READ WRITE AS #1
 PUT #1, 13, INITDAT
 CLOSE #1
END IF

END SUB
```

APPENDIX B

CSSDISP PROGRAM LISTINGS

```
DEFINT A-Z

'-----------------------------------------------

'===============================================
' Program Name: CSSDISP
'
' File Name: CSSDISP.BAS
'
' Description: This routine is the display module for the colloidal
' suspension simulator.
'
' Author: Peter T. Robinson
'
' Date: January 24, 1992
'
' Inputs: None
'
' Outputs: None
'
' Revision History:
' None
'
'===============================================
'
DECLARE SUB DISPINFO ()
DECLARE SUB DISPLAYBOX ()
DECLARE SUB DISPLAY (X() AS DOUBLE, Y() AS DOUBLE, Z() AS DOUBLE,
X0() AS DOUBLE, Y0() AS DOUBLE, Z0() AS DOUBLE, RADIUS() AS DOUBLE,
COLR())
DECLARE SUB TRANSDATA ()
DECLARE SUB READSTATE0 (FILENAME$, X() AS DOUBLE, Y() AS DOUBLE,
RADIUS() AS DOUBLE, COLR() AS INTEGER, X0() AS DOUBLE, Y0() AS DOU-
BLE)
DECLARE SUB CLEARINFOBOX ()
DECLARE SUB DISPINFOBOX ()
DECLARE SUB LOADDATA (FILEPAR$)
DECLARE SUB GETFILENAME (FILENAME$, FILEPAR$)
DECLARE SUB DISPMAIN ()

DIM SHARED DOT AS INTEGER

'$INCLUDE: 'CSSCOM.INC'

ON ERROR GOTO ErrorHandler

DOT = -1

'===============================
' Display the Main Screen
'===============================

CALL DISPMAIN

'===============================
' Obtain the filename where to read the data.
'===============================

CALL GETFILENAME(FILENAME$, FILEPAR$)
IF FILENAME$ = "QUIT" THEN RUN CSSMENU$

'=========================
' Load the parameter file.
'=========================

CALL LOADDATA(FILEPAR$)

'=================================
' Determine the number of particles
'=================================

IF Simulation = 1 THEN
 NUMPART = PartInfoA.Number + PartInfoB.Number
ELSEIF Simulation = 2 THEN
 NUMPART = PartInfoA.Number
END IF

DIM X(1 TO NUMPART) AS DOUBLE ' X position
```

```
DIM Y(1 TO NUMPART) AS DOUBLE ' Y postion
DIM Z(1 TO NUMPART) AS DOUBLE ' Z postion

DIM X0(1 TO NUMPART) AS DOUBLE ' X position
DIM Y0(1 TO NUMPART) AS DOUBLE ' Y postion
DIM Z0(1 TO NUMPART) AS DOUBLE ' Z postion

DIM XOLD(1 TO NUMPART) AS DOUBLE ' Old X position
DIM YOLD(1 TO NUMPART) AS DOUBLE ' Old Y postion
DIM ZOLD(1 TO NUMPART) AS DOUBLE ' Old Z postion

DIM RADIUS(1 TO NUMPART) AS DOUBLE ' Particle color
DIM COLR(1 TO NUMPART) AS INTEGER ' Particle color

'=============================================================
' Define constants and initialize variables.
'=============================================================

REPLSPD& = 20
UPARROW = 72
DOWNARROW = 80
ESCAPE = 27
ENTER = 13
SPACE = 32

Pi# = 3.141592654# ' The value of Pi
T# = TimeIncr

AGAIN = 0 ' Equal to 0 if do not run again.
 ' Equal to 1 if run again.
DONE = 0
FIRST = 1
CNT = 1

XMIN = 0 ' Minimum X screen coordinate.
XMAX = 510 ' Maximum X screen coordinate.
YMIN = 20 ' Minimum Y screen coordinate.
YMAX = 410 ' Maximum Y screen coordinate.

BLACK = 0
BLUE = 1
GREEN = 2
YELLOW = 14
WHITE = 15

BACKCOLR = BLUE
TITLE = YELLOW
BACK = BLUE

'=============================================================
' Begin the major loop for displaying the dynamics of the
' coloidal system for specified conditions.
'=============================================================

DO ' Major Loop 1

'=============================================================
' Initialize the colloidal system
'=============================================================

REPLFLAG = 0
COUNT = 0
MTIME = 0
TLIMIT = 0

'=============================================================
' Draw the graphics window.
'=============================================================

SCREEN 12
CLS

PAINT (XMAX - 10, YMAX - 10), BLACK, 1
LINE (XMIN, YMIN)-(XMAX, YMAX), BACK, BF
LINE (XMIN, YMIN)-(XMAX, YMAX), WHITE, B
LOCATE 1, 16: COLOR TITLE: PRINT "Colloidal Suspension Simulator"
LINE (100, 0)-(375, 15), WHITE, B
```

```
'=========================================================
' Display the screen data.
'=========================================================

CALL DISPINFO

IF FIRST = 1 THEN
LOCATE 28, 2
COLOR 14 'YELLOW
PRINT " Press any key to Start the simulation "

DO
LOOP WHILE INKEY$ = ""

LOCATE 28, 2
PRINT " "
FIRST = 0
END IF

'=========================================================
' Display the initial positions of the particles.
'=========================================================

CALL READSTATE0(FILENAME$, X(), Y(), RADIUS(), COLR(), X0(), Y0())

CALL DISPLAY(X(), Y(), Z(), X0(), Y0(), Z0(), RADIUS(), COLR())


'=========================================================
' Begin the loop to display each state of the colloidal
' system at each time increment.
'=========================================================

DO WHILE NOT EOF(2) AND PRESS$ = "" OR UCASE$(PRESS$) =
CHR$(ENTER)

'=========================================================
' Update the counter on the display.
'=========================================================

TLIMIT = TLIMIT + 1
MTIME = MTIME + 1
IF MTIME > 9999 THEN TIME = 0

COUNT = COUNT + 1

'=========================================================
' Pause if the <ENTER> key has been pressed.
'=========================================================

LOCATE 28, 24
COLOR 14
PRINT "Press <ENTER> to Pause"

PRESS$ = INKEY$

STALL& = (20 - REPLSPD&) ^ 2 * 100
M& = 0
DO
PRESS2$ = INKEY$
M& = M& + 1
LOOP WHILE M& <= STALL& AND PRESS2$ = ""
IF PRESS2$ <> "" THEN PRESS$ = PRESS2$

IF UCASE$(PRESS$) = CHR$(ENTER) THEN

LOCATE 28, 2
COLOR 14
PRINT "Pause"
DO
LOOP WHILE INKEY$ = ""

LOCATE 28, 2
PRINT " "
PRESS$ = ""
END IF

' ELSEIF REPLSPD& < 20 THEN
```

```
' END IF

'=========================================================
' Display the new positions of the particles.
'=========================================================

INPUT #2, FRAMENUM

FOR I = 1 TO NUMPART
X0(I) = X(I)
Y0(I) = Y(I)
INPUT #2, X(I), Y(I)
NEXT I

CALL DISPLAY(X(), Y(), Z(), X0(), Y0(), Z0(), RADIUS(), COLR())

LOOP

CLOSE #2

'=========================================================
' Determine what to do if a key has been pressed.
'=========================================================

PRESS$ = INKEY$

LOCATE 1, 55
COLOR 14
PRINT "Replay Speed: ";
COLOR 15
PRINT USING "##"; REPLSPD&

LOCATE 28, 2
COLOR 14
PRINT " Press <SPACE BAR> to Replay, Use Arrows to adjust Speed, Press <ESC>
to Exit "
DO

BUTTON$ = UCASE$(INKEY$)
IF BUTTON$ <> "" THEN ' Test for key press.

Ky = ASC(RIGHT$(BUTTON$, 1))

SELECT CASE Ky
CASE ESCAPE
DONE = 1
' END
CASE 68 'D
IF Ky = 68 THEN DOT = DOT * -1
CASE UPARROW
IF Ky = UPARROW THEN
REPLSPD& = REPLSPD& + 1
IF REPLSPD& > 20 THEN
REPLSPD& = 20
BEEP
END IF
LOCATE 1, 55
COLOR 14
PRINT "Replay Speed: ";
LOCATE 1, 69
COLOR 15
PRINT USING "##"; REPLSPD&
BUTTON$ = ""
END IF
CASE DOWNARROW
IF Ky = DOWNARROW THEN
REPLSPD& = REPLSPD& - 1
IF REPLSPD& < 0 THEN
REPLSPD& = 0
BEEP
END IF
LOCATE 1, 55
COLOR 14
PRINT "Replay Speed: ";
LOCATE 1, 69
COLOR 15
PRINT USING "##"; REPLSPD&
```

```
BUTTON$ = ""
END IF
END SELECT
END IF
LOOP WHILE BUTTON$ <> CHR$(SPACE) AND BUTTON$ <> CHR$(ESCAPE)

LOOP UNTIL DONE = 1 ' Major Loop 1

'=========================================================
' Return to the CSS.EXE program.
'=========================================================

RUN CSSMENU$

'=========================================================
' Check for what error has occurred
'=========================================================

ErrorHandler:

SCREEN 0
SCREEN 12
PRINT " "
PRINT " "
PRINT " "
PRINT " "
PRINT " "
PRINT " An Error has occured - Press any key to continue."
DO
LOOP WHILE INKEY$ = ""
RUN CSSMENU$

SUB CLEARINFOBOX

'-----------------------------------------------------------

'=========================================================
' Program Name: CLEARINFOBOX
'
' Description: This subroutine clears the message at the
' bottom of the screen.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=========================================================
'

'$INCLUDE: 'MENUPAR.INC'

COLOR WHITE, CYAN

LCOL% = 4
TROW% = 21
BROW% = 24
RCOL% = 76
LABEL$ = ""
FORE% = CYAN
BACK% = CYAN
PAGE% = 0
FRAME% = 1
TYP% = 0

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

END SUB

'
'=========================================================
'
'=========================================================
SUB DISPINFO
```

```
LWHITE = 15
YELLOW = 14
COLOR LWHITE


LOCATE 27, 3
PRINT PartInfoA.ComName
CIRCLE (5, 422), 4, 15

IF Simulation = 1 THEN
 LOCATE 27, 26
 PRINT PartInfoB.ComName
 CIRCLE (188, 422), 4, 10
ELSEIF Simulation = 2 THEN
 LOCATE 27, 26
 PRINT FiberInfo.ComName
 LINE (184, 418)-(192, 426), 10, BF
END IF

LOCATE 27, 49
PRINT MediumInfo1.ComName
LINE (367, 419)-(375, 427), 1, BF

LOCATE 3, 66
COLOR LWHITE
PRINT "Model:";
LOCATE 3, 73
COLOR YELLOW
IF DYNMDL = 1 THEN
 PRINT "DLVO"
ELSEIF DYNMDL = 2 THEN
 PRINT "Acid/Bs"
ELSEIF DYNMDL = 3 THEN
 PRINT "Random"
END IF

LOCATE 5, 66
COLOR LWHITE
PRINT "pH Level: ";
'LOCATE 5, 68
COLOR YELLOW
PRINT USING "##.#"; pHInfo1

LOCATE 7, 66
COLOR LWHITE
PRINT "Zeta(Part. A): ";
LOCATE 8, 68
COLOR YELLOW
PRINT USING "###.##"; ZetaInfoA;
PRINT " mV"

IF Simulation = 1 THEN
 LOCATE 10, 66
 COLOR LWHITE
 PRINT "Zeta(Part. B): ";
 LOCATE 11, 68
 COLOR YELLOW
 PRINT USING "###.##"; ZetaInfoB;
 PRINT " mV"
ELSEIF Simulation = 2 THEN
 LOCATE 10, 66
 COLOR LWHITE
 PRINT "Zeta(Fiber): ";
 LOCATE 11, 68
 COLOR YELLOW
 PRINT USING "###.##"; ZetaInfoF;
 PRINT " mV"
END IF

LOCATE 13, 66
COLOR LWHITE
PRINT "Debye Length: "
LOCATE 14, 66
COLOR YELLOW
'PRINT USING "#####.##"; K
PRINT USING "##.##^^^^"; K;
PRINT " 1/m"

LOCATE 16, 66
```

```
COLOR LWHITE
PRINT "Concentration: "
LOCATE 17, 66
COLOR YELLOW
'PRINT USING "##.######"; ElectroInfo1.Concen
PRINT USING "##.##^^^^"; ElectroInfo1.Concen;
PRINT " N"

LOCATE 19, 66
COLOR LWHITE
PRINT "Time Increment:"
LOCATE 20, 66
COLOR YELLOW
'PRINT USING "##.#######"; TimeIncr
PRINT USING "##.##^^^^"; TimeIncr;
PRINT " sec"

LOCATE 22, 66
COLOR LWHITE
PRINT "Frame Number:"
LOCATE 23, 69
COLOR YELLOW
PRINT USING "######"; MTIME

COLOR 15

LINE (515, 20)-(639, 410), 15, B

END SUB

SUB DISPINFOBOX

'-------------------------------------------------------

'=======================================================
' Program Name: DISPINFOBOX
'
' Description: This subroutine displays a message at the
' bottom of the screen.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=======================================================
'

'$INCLUDE: 'MENUPAR.INC'

COLOR WHITE, CYAN

LCOL% = 4
TROW% = 21
BROW% = 24
RCOL% = 76
LABEL$ = ""
FORE% = WHITE
BACK% = CYAN
PAGE% = 0
FRAME% = 1
TYP% = 0

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

END SUB

'
'=======================================================
'
'=======================================================
SUB DISPLAY (X() AS DOUBLE, Y() AS DOUBLE, Z() AS DOUBLE, X0() AS
DOUBLE, Y0() AS DOUBLE, Z0() AS DOUBLE, RADIUS() AS DOUBLE, COLR())

'=======================================================
```

```
' Initialize values.
'=============================================================

XMIN = 0
XMAX = 510
YMIN = 20
YMAX = 410

'XREAL! = 60! ' 60 microns
'YREAL! = 40! ' 40 microns
XREAL! = 150! ' 150 microns
YREAL! = 100! ' 100 microns

WHITE = 15
YELLOW = 14
GREEN = 10
BLACK = 0
BLUE = 1

TITLE = YELLOW
BACK = BLUE

'=============================================================
' Begin the major loop to display each particle.
'=============================================================

FOR I = 1 TO NUMPART

'=============================================================
' Draw the box.
'=============================================================

 LINE (XMIN, YMIN)-(XMAX, YMAX), WHITE, B

 IF Simulation = 2 THEN
'=============================================================
' Draw the fiber.
'=============================================================
 LINE (2, YMIN + 2)-(7, YMAX - 2), GREEN, BF
 END IF

'=============================================================
' Convert the real coordinates and radii values to
' screen coordinates.
'=============================================================

' RAD# = (RADIUS(I) * 10000!) * ((XMAX - XMIN) / XREAL!)
 RAD# = (RADIUS(I)) * ((XMAX - XMIN) / XREAL!)

 XOLD# = X0(I) * ((XMAX - XMIN) / XREAL!) + XMIN
 YOLD# = 400! - Y0(I) * ((YMAX - YMIN) / YREAL!) + YMIN

 XNEW# = X(I) * ((XMAX - XMIN) / XREAL!) + XMIN
 YNEW# = 400! - Y(I) * ((YMAX - YMIN) / YREAL!) + YMIN

'=============================================================
' Plot a circle with the back ground color
' if the X and Y positions are on the screen.
'=============================================================

 IF ((XOLD# - RAD# > XMIN AND XOLD# + RAD# < XMAX) AND (YOLD# -
RAD# > YMIN AND YOLD# + RAD# < YMAX)) THEN
 CIRCLE (XOLD#, YOLD#), RAD#, BACK
 END IF

'=============================================================
' Plot a circle with the particle color
' if the X and Y positions are on the screen.
'=============================================================

' IF ((XNEW# > XMIN AND XNEW# < XMAX) AND (YNEW# > YMIN AND
YNEW# < YMAX)) THEN
 IF ((XNEW# - RAD# > XMIN AND XNEW# + RAD# < XMAX) AND (YNEW# -
RAD# > YMIN AND YNEW# + RAD# < YMAX)) THEN
 CIRCLE (XNEW#, YNEW#), RAD#, COLR(I)
 IF DOT = 1 THEN
 PSET (XNEW#, YNEW#), COLR(I)
 END IF
```

```
END IF

NEXT I

'=============================================================
' Update the experiment time.
'=============================================================

LOCATE 23, 69
COLOR 14
PRINT USING "######"; MTIME
COLOR 15

END SUB

SUB DISPMAIN

'-------------------------------------------------------

'=============================================================
' Program Name: DISPMAIN
'
' Description: This subroutine displays the main menu.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'
'$INCLUDE: 'MENUPAR.INC'

FORGND = WHITE
BACGND = BLUE
FORGND2 = WHITE
BACGND2 = CYAN

HIGHLT = YELLOW

SCREEN 0

COLOR WHITE, CYAN
CLS

LCOL% = 5
TROW% = 2
BROW% = 17
RCOL% = 78
LABEL$ = "Colloidal Suspension Simulator - Display Model"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1
TYP% = 2

'CLS

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

END SUB

SUB GETFILENAME (FILENAME$, FILEPAR$)

'-------------------------------------------------------

'=============================================================
' Program Name: GETFILENAME
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
```

```
'
' Revision History:
' None
'
'=============================================================
'
'$INCLUDE: 'MENUPAR.INC'

DONE = 0
DO

 CALL DISPINFOBOX
 BUFFER$ = ""
 LOCATE 22, 5
 COLOR YELLOW, CYAN
 PRINT "Enter The Input File Name (With no File Extension): ";
 COLOR WHITE, CYAN
 LOCATE 22, 58
 INPUT ; "", BUFFER$
 FILENAME$ = BUFFER$
 CALL CLEARINFOBOX

 IF LEN(FILENAME$) > 8 THEN
 MESSAGE$ = " Invalid File Name"
 LOCATE 20, 5
 PRINT MESSAGE$
 MESSAGE$ = " File names must be eight characters or less."
 LOCATE 21, 5
 PRINT MESSAGE$
 FILENAME$ = ""
 END IF

 IF FILENAME$ <> "" THEN
 IF FILENAME$ = "QUIT" THEN STOP
 FILEPAR$ = FILENAME$ + ".PAR"
 FILENAME$ = FILENAME$ + ".DAT"
 FILEP$ = FILEPAR$ + CHR$(0)
 CALL EXIST(FILEP$, FILEXISTS%)
 IF FILEXISTS% THEN
 FILEN$ = FILENAME$ + CHR$(0)
 CALL EXIST(FILEN$, FILEXIST%)
 IF FILEXIST% THEN
 DONE = 1
 ELSE
 LOCATE 23, 18
 COLOR YELLOW, CYAN
 PRINT "No Data File found for "; UCASE$(FILEN$)
 DO
 LOOP WHILE INKEY$ = ""
 END IF
 ELSE
 COLOR YELLOW, CYAN
 LOCATE 23, 20
 PRINT "The File: "; UCASE$(FILEP$); " Does not exist"
 DO
 LOOP WHILE INKEY$ = ""
 END IF

 ELSE
 COLOR YELLOW, CYAN
 R$ = "N"
 DO
 LOCATE 23, 10
 INPUT "Do You Wish To Quit? [N]: "; R$
 LOOP WHILE UCASE$(R$) = "Y" AND UCASE$(R$) = "N"
 IF UCASE$(R$) = "Y" THEN
 DONE = 1
 FILENAME$ = "QUIT"
 END IF
 LOCATE 22, 50
 PRINT " ";
 LOCATE 23, 10
 PRINT " ";
 END IF
LOOP WHILE (DONE = 0)


END SUB
```

```
SUB LOADDATA (FILEPAR$)

IF FILEPAR$ <> "" THEN

FILEN$ = FILEPAR$ + CHR$(0)
CALL EXIST(FILEN$, FILEXISTS%)

IF FILEXISTS% THEN
OPEN FILEPAR$ FOR RANDOM ACCESS READ WRITE AS #1

GET #1, 1, PartInfoA
GET #1, 2, PartInfoB
GET #1, 3, MediumInfo1
GET #1, 4, ElectroInfo1
GET #1, 5, FiberInfo
GET #1, 6, MENULVL
GET #1, 7, pHInfo1
GET #1, 8, ZetaInfoA
GET #1, 9, ZetaInfoB
GET #1, 10, ZetaInfoF
GET #1, 11, DYNMDL
GET #1, 12, Simulation
GET #1, 13, INITDAT
GET #1, 14, TimeIncr
GET #1, 15, K

CLOSE #1

ELSE
FOR LL = 1 TO 19
' MESSAGE$(LL) = " "
NEXT LL

' MESSAGE$(6) = " The file: " + FILENAME$
' MESSAGE$(7) = " Does not exist"
' CALL DISPERR(MESSAGE$())
END IF
END IF

END SUB

SUB READSTATE0 (FILENAME$, X() AS DOUBLE, Y() AS DOUBLE, RADIUS()
AS DOUBLE, COLR() AS INTEGER, X0() AS DOUBLE, Y0() AS DOUBLE)

OPEN FILENAME$ FOR INPUT AS #2

INPUT #2, FILETYPE$
INPUT #2, NPART

FOR I = 1 TO NPART
 INPUT #2, RADIUS(I), COLR(I)
NEXT I

INPUT #2, FRAMENUM

FOR I = 1 TO NPART
 INPUT #2, X(I), Y(I)
' X0(I) = X(I)
' Y0(I) = Y(I)
NEXT I

END SUB

'
'============================================================
'
'
'============================================================
'
SUB TRANSDATA

'TFIL$ = "C:\QB\CSSM\CSSTEMP"
TFIL$ = "CSSTEMP"
TFILE$ = "CSSTEMP" + CHR$(0)

CALL EXIST(TFILE$, FILEXISTS%)
```

```
IF FILEXISTS% THEN

 OPEN TFIL$ FOR RANDOM ACCESS READ WRITE AS #1

 GET #1, 1, PartInfoA
 GET #1, 2, PartInfoB
 GET #1, 3, MediumInfo1
 GET #1, 4, ElectroInfo1
 GET #1, 5, FiberInfo
 GET #1, 6, MENULVL
 GET #1, 7, pHInfo1
 GET #1, 8, ZetaInfoA
 GET #1, 9, ZetaInfoB
 GET #1, 10, ZetaInfoF
 GET #1, 11, DYNMDL
 GET #1, 12, Simulation
 GET #1, 13, INITDAT
 GET #1, 14, TimeIncr

 CLOSE #1
ELSE
 CLS
 SCREEN 12
 PRINT
 PRINT
 PRINT " CSS.EXE must be run first"
END IF

END SUB
```

APPENDIX C

CSSRUN PROGRAM LISTINGS

```
DEFINT A-Z

'=============================================================
' Program Name: CSSRUN
'
' File Name: CSSRUN.BAS
'
' Description: This routine reads input provided by the
' cssmenu routine and performs the necessary
' calculations used by the cssdisplay routine.
'
' Author: Peter T. Robinson
'
' Date: January 24, 1992
'
' Inputs: None
'
' Outputs: None
'
' Revision History:
' None
'
'=============================================================
'
DECLARE SUB DISPMAIN ()
DECLARE SUB CHECKWALL (X#, Y#, XOLD#, YOLD#)
DECLARE SUB CLEARINFOBOX ()
DECLARE SUB DISPINFOBOX ()
DECLARE SUB EULERV (II, RADIUS() AS DOUBLE, X() AS DOUBLE, Y() AS
DOUBLE, XDOT() AS DOUBLE, YDOT() AS DOUBLE, FX() AS DOUBLE, FY()
AS DOUBLE)
DECLARE SUB GETFXFY (II, RADIUS() AS DOUBLE, X() AS DOUBLE, Y() AS
DOUBLE, XDOT() AS DOUBLE, YDOT() AS DOUBLE, FX() AS DOUBLE, FY()
AS DOUBLE)
DECLARE SUB KAPPA ()
DECLARE FUNCTION PPATRACT# (D#, R1#, R2#)
DECLARE FUNCTION PPREPEL# (D#, R1#, R2#, TYP1, TYP2)
DECLARE SUB TRANSDATA ()
DECLARE SUB TRANSDATAP (FILEPAR$)
DECLARE FUNCTION GAUSS# ()
DECLARE SUB GETFILENAME (FILENAME$, FILEPAR$)
DECLARE SUB PRINTDATA (RECNUM, X() AS DOUBLE,
Y() AS DOUBLE, Z() AS DOUBLE)
DECLARE SUB PRINTHEADER (RECNUM%,
X() AS DOUBLE, Y() AS DOUBLE, Z() AS DOUBLE,
 RADIUS() AS DOUBLE, COLR%())
DECLARE SUB GETTIMELIMIT (TIMELIMIT)
DECLARE FUNCTION SIGN% ()

' $INCLUDE: 'CSSCOM.INC'
' $INCLUDE: 'MENUPAR.INC'

ON ERROR GOTO ErrorHandler
RANDOMIZE (TIMER)

'=============================
' Display the Main Screen
'=============================

CALL DISPMAIN
'=================================
' Obtain the filename where to write out the data.
'=================================
CALL GETFILENAME(FILENAME$, FILEPAR$)
IF FILENAME$ = "QUIT" THEN RUN CSSMENU$
'=================================
' Create the file FILENAME$
'=================================
OPEN FILENAME$ FOR OUTPUT AS #2
'=================================
' Transfer the data that was obtained through CSS.EXE
'=================================
CALL TRANSDATA

'=================================
' Calculate the Debye length,i.e. K
'=================================
```

```
'=============================================================
' Calculate the Debye length,i.e. K
'=============================================================

CALL KAPPA

'=============================================================
' Determine the number of particles
'=============================================================

IF Simulation = 1 THEN
 NUMPART = PartInfoA.Number + PartInfoB.Number
ELSEIF Simulation = 2 THEN
 NUMPART = PartInfoA.Number
END IF

'=============================================================
' Obtain the time limit of the experiment
'=============================================================

CALL GETTIMELIMIT(TIMELIMIT)
IF TIMELIMIT = 0 THEN RUN CSSMENU$

COLOR WHITE, BLUE
LOCATE 10, 50
PRINT "Number of Particles: "
COLOR YELLOW, BLUE
LOCATE 10, 71
PRINT USING "###"; NUMPART

'=============================================================
' Define the arrays that will be used in the calculations
'=============================================================

DIM DENS(1 TO NUMPART) AS DOUBLE ' Particle Density
DIM MASS(1 TO NUMPART) AS DOUBLE ' Particle Mass
DIM RADIUS(1 TO NUMPART) AS DOUBLE ' Particle Radius
DIM VOLUME(1 TO NUMPART) AS DOUBLE ' Particle Volume
DIM COLR(1 TO NUMPART) AS INTEGER ' Particle color

DIM X(1 TO NUMPART) AS DOUBLE ' X position
DIM Y(1 TO NUMPART) AS DOUBLE ' Y position
DIM Z(1 TO NUMPART) AS DOUBLE ' Z position

DIM XDOT(1 TO NUMPART) AS DOUBLE ' Particle velocity; dX/dt
DIM YDOT(1 TO NUMPART) AS DOUBLE ' Particle velocity; dY/dt

DIM DXDT(0 TO 3, 1 TO NUMPART) AS DOUBLE ' The four previous values
 ' of the I differential
 ' equations dX/dt = Vx.
DIM DYDT(0 TO 3, 1 TO NUMPART) AS DOUBLE ' The four previous values
 ' of the I differential
 ' equations dY/dt = Vy.

DIM DXDOTDT(0 TO 3, 1 TO NUMPART) AS DOUBLE 'The four previous values
 ' of the I differential
 ' equations dVx/dt = Fx/m.
DIM DYDOTDT(0 TO 3, 1 TO NUMPART) AS DOUBLE 'The four previous values
 ' of the I differential
 ' equations dVy/dt = Fy/m.

DIM DISTANCE(1 TO NUMPART, 1 TO NUMPART) AS DOUBLE ' Distance
between
 ' particle I and
 ' particle J.

DIM FTTL(1 TO NUMPART, 1 TO NUMPART) AS DOUBLE ' Total force
 ' acting on particle
 ' I due to particle
 ' J

DIM FX(1 TO NUMPART) AS DOUBLE ' The X component of FTTL
DIM FY(1 TO NUMPART) AS DOUBLE ' The Y component of FTTL

NB = 120 ' Number of boxes to prevent
 ' overlap of particles.
```

```
DIM BOXX(1 TO NB) AS DOUBLE ' X positions for the initial
' particles.
DIM BOXY(1 TO NB) AS DOUBLE ' Y positions for the initial
```

```
CNT = 1
XVAL! = 54!
```

```
DIM BOXX(1 TO NB) AS DOUBLE ' X positions for the initial
 ' particles.
DIM BOXY(1 TO NB) AS DOUBLE ' Y positions for the initial
 ' particles.

DIM TAKEN(1 TO NB) ' Equal to 0 if a box is empty.
 ' Equal to 1 if a box is taken.

DIM SHARED Pi#
DIM SHARED STATS
DIM SHARED MinDistvan
DIM SHARED MinDistDbl

'=========================================================
' Define constants and initialize variables.
'=========================================================

MeanA# = PartInfoA.MeanDiameter / 2!
SDevA# = PartInfoA.SDevDiameter / 2!
MeanB# = PartInfoB.MeanDiameter / 2!
SDevB# = PartInfoB.SDevDiameter / 2!

Pi# = 3.141592654# ' The value of Pi
T# = TimeIncr
STATS = -1 ' Show statistics if equal to 1
BROWN# = .5 ' Brownian motion variable
MinDistDbl = 10! ' The minimum distance to calculate the DBL layer force.
MinDistvan = 10! ' The minimum distance to calculate the van der Waals force.

COUNT = 0

FOR I = 1 TO NB
 TAKEN(I) = 0
NEXT I

'=========================================================
' Define the possible coordinates (boxes) where the initial
' particles will be placed.
'=========================================================

CNT = 1
XVAL! = 54!
YVAL! = 39.5 ' Y location of the CNT box.

FOR I = 1 TO 8
 FOR J = 1 TO 15
 BOXX(CNT) = XVAL!
 BOXY(CNT) = YVAL!
 XVAL! = XVAL! + 3
 CNT = CNT + 1
 NEXT J
 XVAL! = 54! ' X location of the CNT box.
 YVAL! = YVAL! + 3
NEXT I

RANDOMIZE (TIMER)

'=========================================================
' For one to the number of particles, initialize
' the radius, density, volume, mass and initial velocity.
'=========================================================

 FOR I = 1 TO NUMPART

 IF I > PartInfoA.Number THEN
 RADIUS(I) = MeanB# + SDevB# * ABS(GAUSS#)
 DENS(I) = PartInfoB.Density
 XDOT(I) = SIGN * RND * PartInfoB.MaxVel
 YDOT(I) = SIGN * RND * PartInfoB.MaxVel
 ELSE
 RADIUS(I) = MeanA# + SDevA# * ABS(GAUSS#)
 DENS(I) = PartInfoA.Density
 XDOT(I) = SIGN * RND * PartInfoA.MaxVel
 YDOT(I) = SIGN * RND * PartInfoA.MaxVel
 END IF

'*********************************************
' Used For Diagnostics
```

```
'
' LOCATE 6, 4
' PRINT USING "####.####"; XDOT(I)
' LOCATE 7, 4
' PRINT USING "####.####"; YDOT(I)
' PRINT GAUSS#
' DO WHILE INKEY$ = "": LOOP
'*********************************************

VOLUME(I) = (4! / 3!) * Pi# * RADIUS(I) ^ 3
CONSTANT# = 1E-12
MASS(I) = DENS(I) * CONSTANT# * VOLUME(I)' Mass in grams

'=============================================================
' Define the initial position and velocity for each
' particle.
'=============================================================

GOOD = 0
DO
PICK = INT(NB * RND) + 1
IF TAKEN(PICK) = 0 THEN GOOD = 1
LOOP WHILE GOOD = 0

TAKEN(PICK) = PICK
X(I) = BOXX(PICK)
Y(I) = BOXY(PICK)

'=============================================================
' Assign initial value conditions
'=============================================================

DXDT(0, I) = XDOT(I)
DYDT(0, I) = YDOT(I)

DXDOTDT(0, I) = 0!
DYDOTDT(0, I) = 0!

IF I <= PartInfoA.Number THEN

COLR(I) = 15 ' WHITE
ELSE
COLR(I) = 10 ' GREEN
END IF

NEXT I

'=============================================================
' Write out the header and the initial position and color
' for the zero state to the file FILENAME$
'=============================================================

LOCATE 3, 12
COLOR WHITE, BLUE
PRINT "The Data is Being Written to the File: "
LOCATE 3, 51
COLOR YELLOW, BLUE
PRINT UCASE$(FILENAME$)

CALL PRINTHEADER(0, X(), Y(), Z(), RADIUS(), COLR())
COLOR WHITE, BLUE
LOCATE 11, 24
PRINT "Current Frame: "
LOCATE 11, 39
COLOR YELLOW, BLUE
PRINT USING "######"; COUNT

'=============================================================
' Begin the loop to calculate each state of the colloidal
' system at each time increment.
'=============================================================

DO

'=============================================================
' Update the counter.
'=============================================================

PRESS$ = UCASE$(INKEY$)
```

```
IF PRESS$ <> "" THEN
Ky = ASC(RIGHT$(PRESS$, 1))
END IF

ESCAPE = 27

COUNT = COUNT + 1

LOCATE 11, 39
PRINT USING "######"; COUNT

FOR II = 1 TO NUMPART

'=========================================================
' Calculate the X and Y force components .
'=========================================================

CALL GETFXFY(II, RADIUS(), X(), Y(), XDOT(), YDOT(), FX(), FY())

COLOR WHITE, BLUE
LOCATE 11, 53
PRINT "Current Particle: "
COLOR YELLOW, BLUE
LOCATE 11, 71
PRINT USING "###"; II

IF COUNT <= 3 THEN

'=========================================================
' Use the Euler modified method to find the
' first three values of position and velocity.
'=========================================================

XOLD# = X(II)
YOLD# = Y(II)

IF DYNMDL = 3 THEN ' Random Motion - only for demonstration purpose.

IF II > PartInfoA.Number THEN
IF PartInfoB.MaxVel > 5! THEN limit = 5!
ELSE
IF PartInfoA.MaxVel > 5! THEN limit = 5!
END IF

X(II) = X(II) + limit * RND * SIGN
Y(II) = Y(II) + limit * RND * SIGN
CALL CHECKWALL(X(II), Y(II), XOLD#, YOLD#)

ELSEIF DYNMDL = 1 THEN ' DLVO Model

X(II) = X(II) + TimeIncr * (XDOT(II) + (TimeIncr / 2!) * (FX(II) / MASS(II)))
Y(II) = Y(II) + TimeIncr * (YDOT(II) + (TimeIncr / 2!) * (FY(II) / MASS(II)))

'=========================================================
' Add Brownian motion
'=========================================================

SGNX = SGN(X(II) - XOLD#)
SGNY = SGN(Y(II) - YOLD#)
IF SGNX = 0 THEN SGNX = SIGN
IF SGNY = 0 THEN SGNY = SIGN

X(II) = X(II) + SGNX * RND * BROWN#
Y(II) = Y(II) + SGNX * RND * BROWN#

CALL CHECKWALL(X(II), Y(II), XOLD#, YOLD#)

END IF

XDOT(II) = XDOT(II) + TimeIncr * ((FX(II) / MASS(II)))
YDOT(II) = YDOT(II) + TimeIncr * ((FY(II) / MASS(II)))

DXDT(COUNT, II) = XDOT(II)
DYDT(COUNT, II) = YDOT(II)

DXDOTDT(COUNT, II) = (FX(II) / MASS(II))
DYDOTDT(COUNT, II) = (FY(II) / MASS(II))
```

```
ELSE

'=========================================================
' Use the fourth order Adams Bashforth method to find the
' new values of position and velocity.
'=========================================================

XOLD# = X(II)
YOLD# = Y(II)

IF DYNMDL = 3 THEN ' Random Motion - only for demonstration purpose.

X(II) = X(II) + 5! * RND * SIGN
Y(II) = Y(II) + 5! * RND * SIGN
CALL CHECKWALL(X(II), Y(II), XOLD#, YOLD#)

ELSEIF DYNMDL = 1 THEN ' DLVO Model

 X(II) = X(II) + (TimeIncr / 2.4) * (5.5 * DXDT(3, II) - 5.9 * DXDT(2, II) + 3.7 *
DXDT(1, II) - .9 * DXDT(0, II))
 Y(II) = Y(II) + (TimeIncr / 2.4) * (5.5 * DYDT(3, II) - 5.9 * DYDT(2, II) + 3.7 *
DYDT(1, II) - .9 * DYDT(0, II))

'=========================================================
' Add Brownian motion
'=========================================================

SGNX = SGN(X(II) - XOLD#)
SGNY = SGN(Y(II) - YOLD#)
IF SGNX = 0 THEN SGNX = SIGN
IF SGNY = 0 THEN SGNY = SIGN

X(II) = X(II) + SGNX * RND * BROWN#
Y(II) = Y(II) + SGNX * RND * BROWN#

CALL CHECKWALL(X(II), Y(II), XOLD#, YOLD#)

END IF

 XDOT(II) = XDOT(II) + (TimeIncr / 2.4) * (5.5 * DXDOTDT(3, II) - 5.9 *
DXDOTDT(2, II) + 3.7 * DXDOTDT(1, II) - .9 * DXDOTDT(0, II))
 YDOT(II) = YDOT(II) + (TimeIncr / 2.4) * (5.5 * DYDOTDT(3, II) - 5.9 *
DYDOTDT(2, II) + 3.7 * DYDOTDT(1, II) - .9 * DYDOTDT(0, II))

'=========================================================
' Preserve the past three values of dX/dt and dV/dt.
'=========================================================

FOR I = 0 TO 2
DXDT(I, II) = DXDT(I + 1, II)
DYDT(I, II) = DYDT(I + 1, II)
DXDOTDT(I, II) = DXDOTDT(I + 1, II)
DYDOTDT(I, II) = DYDOTDT(I + 1, II)
NEXT I

'=========================================================
' Set the new values of dX/dt and dV/dt.
'=========================================================

DXDT(3, II) = XDOT(II)
DYDT(3, II) = YDOT(II)
DXDOTDT(3, II) = (FX(II) / MASS(II))
DYDOTDT(3, II) = (FY(II) / MASS(II))

END IF

NEXT II

'=========================================================
' Write out the new position and color for the new
' state to the file FILENAME$
'=========================================================

CALL PRINTDATA(COUNT, X(), Y(), Z())

LOOP WHILE Ky <> ESCAPE AND COUNT < TIMELIMIT
```

```
'=============================================================
' Close the output data file
'-------------------------------------------------------------
```

```
RUN CSSMENU$
```

```
'=============================================================
' Close the output data file
'=============================================================

CLOSE #2

'=============================================================
' Transfer the parameters to the output file
'=============================================================

CALL TRANSDATAP(FILEPAR$)

COLOR WHITE, BLUE


IF Ky = ESCAPE THEN
 BEEP
 LOCATE 5, 23
 PRINT " Program Terminated Prematurely"
 LOCATE 7, 23
 PRINT "-----> Press C to continue <-----"
 DO
 LOOP WHILE UCASE$(INKEY$) <> "C"
ELSE
 BEEP
 BEEP
 BEEP
 LOCATE 5, 23
 COLOR YELLOW, BLUE
 PRINT "The Program Completed Successfully"
 LOCATE 7, 23
 PRINT "-----> Press C to continue <-----"
 DO
 LOOP WHILE UCASE$(INKEY$) <> "C"
END IF

'=============================================================
' Return to the css menu program.
'=============================================================

RUN CSSMENU$

'
'=============================================================
' Check for what error has occurred
'=============================================================

ErrorHandler:

SCREEN 0
SCREEN 12
PRINT " "
PRINT " "
PRINT " "
PRINT " "
PRINT " "
PRINT " An Error has occured - Press any key to continue."
DO
LOOP WHILE INKEY$ = ""

RUN CSSMENU$

SUB CHECKWALL (X#, Y#, XOLD#, YOLD#)

'Change X direction if particle hits top or bottom edge.
 IF (X# < 1 OR X# > 150) THEN
 X# = XOLD#
 END IF
'Change Y direction if particle hits top or bottom edge.
 IF Y# < 1 OR Y# > 100 THEN
 Y# = YOLD#
 END IF

END SUB

SUB CLEARINFOBOX

'-------------------------------------------------------------
```

```
'=============================================================
' Program Name: CLEARINFOBOX
'
' Description: This subroutine clears the message at the
' bottom of the screen.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

'$INCLUDE: 'MENUPAR.INC'

COLOR WHITE, CYAN

LCOL% = 4
TROW% = 21
BROW% = 24
RCOL% = 76
LABEL$ = ""
FORE% = CYAN
BACK% = CYAN
PAGE% = 0
FRAME% = 1
TYP% = 0

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

END SUB

SUB DISPINFOBOX
```

```
'-------------------------------------------------------

'=============================================================
' Program Name: DISPINFOBOX
'
' Description: This subroutine displays a message at the
' bottom of the screen.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=============================================================
'

'$INCLUDE: 'MENUPAR.INC'

COLOR WHITE, CYAN

LCOL% = 4
TROW% = 21
BROW% = 24
RCOL% = 76
LABEL$ = ""
FORE% = WHITE
BACK% = CYAN
PAGE% = 0
FRAME% = 1
TYP% = 0

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

END SUB

SUB DISPMAIN
```

```
'-----------------------------------------------------

'=====================================================
' Program Name: DISPMAIN
'
' Description: This subroutine displays the main menu.
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'
'$INCLUDE: 'MENUPAR.INC'

FORGND = WHITE
BACGND = BLUE
FORGND2 = WHITE
BACGND2 = CYAN

HIGHLT = YELLOW

SCREEN 0

COLOR WHITE, CYAN
CLS

LCOL% = 5
TROW% = 2
BROW% = 17
RCOL% = 78
LABEL$ = "Colloidal Suspension Simulator - Run Model"
FORE% = WHITE
BACK% = BLUE
PAGE% = 0
FRAME% = 1
TYP% = 2

'CLS

CALL MAKEWINDOW(LCOL%, TROW%, RCOL%, BROW%, LABEL$,
FRAME%, TYP%, FORE%, BACK%, PAGE%)

END SUB

DEFSNG A-Z
FUNCTION GAUSS#

 U1# = RND
 U2# = RND

 GAUSS# = SQR(-2! * LOG(U1#)) * COS(2! * Pi# * U2#)

END FUNCTION

DEFINT A-Z
SUB GETFILENAME (FILENAME$, FILEPAR$)

'-----------------------------------------------------

'=====================================================
' Program Name: GETFILNAME
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=====================================================
'
```

```
'$INCLUDE: 'MENUPAR.INC'

DONE = 0
DO

 CALL DISPINFOBOX
 BUFFER$ = ""
 LOCATE 22, 5
 COLOR YELLOW, CYAN
 PRINT "Enter The Output File Name (With no File Extension): ";
 COLOR WHITE, CYAN
 LOCATE 22, 58
 INPUT ; "", BUFFER$
 FILENAME$ = BUFFER$
 CALL CLEARINFOBOX

 IF LEN(FILENAME$) > 8 THEN
 MESSAGE$ = " Invalid File Name"
 LOCATE 20, 5
 PRINT MESSAGE$
 MESSAGE$ = " File names must be eight characters or less."
 LOCATE 21, 5
 PRINT MESSAGE$
 FILENAME$ = ""
 END IF

 IF FILENAME$ <> "" THEN
 IF FILENAME$ = "QUIT" THEN STOP
 FILEPAR$ = FILENAME$ + ".PAR"
 FILENAME$ = FILENAME$ + ".DAT"
 DONE = 1
 ELSE
 COLOR YELLOW, CYAN
 R$ = "N"
 DO
 LOCATE 23, 10
 INPUT "Do You Wish To Quit? [N]: "; R$
 LOOP WHILE UCASE$(R$) = "Y" AND UCASE$(R$) = "N"
 IF UCASE$(R$) = "Y" THEN
 DONE = 1
 FILENAME$ = "QUIT"
 END IF
 LOCATE 22, 50
 PRINT " ";
 LOCATE 23, 10
 PRINT " ";
 END IF
LOOP WHILE (DONE = 0)

END SUB

SUB GETFXFY (II, RADIUS() AS DOUBLE, X() AS DOUBLE, Y() AS DOUBLE,
XDOT() AS DOUBLE, YDOT() AS DOUBLE, FX() AS DOUBLE, FY() AS DOU-
BLE)

'-------------------------------------------------------

'=======================================================
' Program Name: GETFXFY
'
' Description:
'
' Author: Peter T. Robinson
'
' Date:
'
' Inputs: None
'
' Outputs: None
'
' Revision History:
' None
'
'=======================================================
'

'=======================================================
```

```
FOR JJ = 1 TO NUMPART
 FX(JJ) = 0!
 FY(JJ) = 0!
NEXT JJ

FOR JJ = 1 TO NUMPART

 IF II <> JJ THEN

'=========================================================
' Calculate the total distance between particle I
' and particle J.
'=========================================================

 XDIFF2# = (X(JJ) - X(II)) ^ 2
 YDIFF2# = (Y(JJ) - Y(II)) ^ 2

 DIST# = (XDIFF2# + YDIFF2#) ^ .5

'=========================================================
' Calculate the total force between particle I and
' particle J and the force between particle I and
' the fiber if running Simulation 2.
'=========================================================

 D# = DIST#
 R1# = RADIUS(II)
 R2# = RADIUS(JJ)

'=========================================================
' Calculate the London van der Waals force between
' two spherical particles.
'=========================================================

 IF DIST# < MinDistvan AND DIST# > 0! THEN
 FATRACTPP# = PPATRACT#(D#, R1#, R2#)
 ELSE
 FATRACTPP# = 0!
 END IF

 D# = DIST#
 R1# = RADIUS(II)
 R2# = RADIUS(JJ)

 IF II <= PartInfoA.Number THEN
 TYP1 = 1
 DENTY# = PartInfoA.Density
 ELSE
 TYP1 = 2
 DENTY# = PartInfoB.Density
 END IF

 IF JJ <= PartInfoA.Number THEN
 TYP2 = 1
 ELSE
 TYP2 = 2
 END IF

'=========================================================
' Calculate the double layer repulsive force between
' two spherical particles.
'=========================================================

 IF D# < MinDistDbl AND D# > 0! THEN
 FREPELPP# = PPREPEL#(D#, R1#, R2#, TYP1, TYP2)
 ELSE
 FREPELPP# = 0!
 END IF

 ' IF SIMULATION = 2 THEN
 ' IF FIBERCNT <= NUMPART THEN
 '
 ' DPF# = X(FIBERCNT)
 ' R1# = RADIUS(FIBERCNT)
 '
'=========================================================
' Calculate the London van der Waals force between
' a spherical particle and a flat plane (fiber).
```

```
'=============================================================
'
' FATRACTPF# = PFATRACT#(D#, R1#)
'
' DPF# = X(FIBERCNT)
' R1# = RADIUS(FIBERCNT)
'
'=============================================================
' Calculate the double layer force between
' a spherical particle and a flat plane (fiber).
'=============================================================
'
' FREPELPF# = PFREPEL#(D#, R1#)
'
' FIBERCNT = FIBERCNT + 1
' ELSE
'
' FATRACTPF# = 0!
' FREPELPF# = 0!
' END IF
' END IF
'
' IF SIMULATION = 1 THEN
' FTOTAL# = FATRACTPP# + FREPELPP#
' ELSE
' FTOTAL# = FATRACTPP# + FREPELPP# + FATTRACTFP# + FREPELPF#
' END IF

'=============================================================
' Calculate the total force acting on particle I due to
' particle J.
'=============================================================

FTOTAL# = FATRACTPP# + FREPELPP#

'=============================================================
' Calculate the X and Y force components on particle II
'=============================================================
FX(II) = FX(II) + FTOTAL# * (X(JJ) - X(II)) / DIST#
FY(II) = FY(II) + FTOTAL# * (Y(JJ) - Y(II)) / DIST#

END IF

NEXT JJ

'=============================================================
' Add the force due to gravity
'=============================================================

C4D3# = 4! / 3!
GCONST# = 980.665 'gm cm/sec^2
RADI# = RADIUS(II)
CONVFAC# = 10000! ^ 3 'Convert 1/cm^2 to 1/um^2

FGRAV# = -(C4D3# * Pi# * (RADI# ^ 3) * (DENTY# - MediumInfo1.Density) *
GCONST#) / CONVFAC#
FY(II) = FY(II) + FGRAV#

'=============================================================
' Add the Stokes law viscos drag force
'=============================================================

VISC# = MediumInfo1.Viscosity

FSTOKEX# = -(6! * Pi# * VISC# * RADI# * XDOT(II)) / 10000! ^ 2
FSTOKEY# = -(6! * Pi# * VISC# * RADI# * YDOT(II)) / 10000! ^ 2

FX(II) = FX(II) + FSTOKEX#
FY(II) = FY(II) + FSTOKEY#

'=============================================================

' ********* Used for diagnosis ******
BUTTON$ = UCASE$(INKEY$)
IF BUTTON$ <> "" THEN ' Test for key press.
 Ky2 = ASC(RIGHT$(BUTTON$, 1))
END IF
```

LOCATE LL, 10
PRINT WIPE$

```
IF Ky2 = 70 THEN 'The letter F
STATS = STATS * -1
END IF

IF STATS = 1 THEN
 'LOCATE 10, 5
 'PRINT DIST#
 WHITE = 15
 YELLOW = 14
 COLOR WHITE
 LOCATE 13, 19
 PRINT "Double Layer Force: ";
 COLOR YELLOW
 PRINT USING "###.###^^^^"; FREPELPP#
 LOCATE 14, 18
 COLOR WHITE
 PRINT "van der Waals Force: ";
 COLOR YELLOW
 PRINT USING "###.###^^^^"; FATRACTPP#
 LOCATE 15, 30
 COLOR WHITE
 PRINT "Gravity: ";
 COLOR YELLOW
 PRINT USING "###.####^^^^"; FGRAV#
 LOCATE 16, 30
 COLOR WHITE
 PRINT "Drag Fx: ";
 COLOR YELLOW
 PRINT USING "###.####^^^^"; FSTOKEX#
 LOCATE 17, 30
 COLOR WHITE
 PRINT "Drag Fy: ";
 COLOR YELLOW
 PRINT USING "###.####^^^^"; FSTOKEY#
ELSE
 WIPE$ = " "
 FOR LL = 13 TO 17
 COLOR 15, 1
 LOCATE LL, 10
 PRINT WIPE$
 NEXT LL
END IF
IF Ky2 = 32 THEN 'Space Bar
 DO WHILE INKEY$ = "": LOOP
END IF

'=================================================

 FX(II) = FX(II) * 10000' [(gm cm)/sec^2] to [(gm um)/sec^2]
 FY(II) = FY(II) * 10000' [(gm cm)/sec^2] to [(gm um)/sec^2]

END SUB

SUB GETTIMELIMIT (TIMELIMIT)

'---------------------------------------------------------

'=================================================
' Program Name: GETTIMELIMIT
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=================================================
'
'$INCLUDE: 'MENUPAR.INC'

 VALID = 0
 TEMPBUF$ = ""
 CALL DISPINFOBOX
```

```
BUFFER$ = ""

DO
BUFFER$ = ""
COLOR YELLOW, CYAN
LOCATE 22, 5
PRINT "Enter The Number of Frames to be Calculated: ";
COLOR WHITE, CYAN
LOCATE 22, 50
INPUT ; "", BUFFER$
TEMPBUF$ = BUFFER$
' IF BUFFER$ <> "" THEN

BUFFER! = VAL(BUFFER$)

IF BUFFER! > 0 AND BUFFER! <= 30000 THEN
TIMELIMIT = BUFFER!
VALID = 1
ELSEIF BUFFER! = 0 OR BUFFER$ = "" THEN
COLOR YELLOW, CYAN
R$ = "N"
DO
LOCATE 23, 10
INPUT "Do You Wish To Quit? [N]: "; R$
LOOP WHILE UCASE$(R$) = "Y" AND UCASE$(R$) = "N"
IF UCASE$(R$) = "Y" THEN
VALID = 1
QUIT = 1
END IF
LOCATE 22, 50
PRINT " ";
LOCATE 23, 10
PRINT " ";
ELSE
LOCATE 23, 10
COLOR YELLOW, CYAN
PRINT "Valid input is an integer number between 1 and 30,000"
LOCATE 22, 50
PRINT " ";
END IF
' ELSE
' VALID = 1
' END IF
LOOP UNTIL VALID = 1

IF QUIT <> 1 THEN
COLOR WHITE, BLUE
LOCATE 10, 5
PRINT "Number of Frames to be Generated: "
LOCATE 10, 39
COLOR YELLOW, BLUE
PRINT USING "######"; TIMELIMIT
CALL CLEARINFOBOX
END IF

END SUB

SUB KAPPA

'----------------------------------------------

'==============================================
' Program Name: KAPPA
'
' Description:
'
' Author: Peter T. Robinson
'
' Date:
'
' Inputs: None
'
' Outputs: None
'
' Revision History:
' None
'
'==============================================
```

```
'
q# = 1.60217733D-19 ' Coulomb
NA# = 6.0221367D+23 ' 1/mole
PermFree# = 8.854187799999999D-12 ' Farad/meter
Bk# = 1.380658E-23 ' Joule/Kelvin

TMP# = (MediumInfo1.Temp) + 273.15 'Kelvin
Dielec# = (MediumInfo1.Perm) * PermFree#

NUMER# = (2!) * (q# ^ 2) * (1000!) * (NA#) * (ElectroInfo1.Concen)

DENOM# = (Dielec#) * (Bk#) * (TMP#)

K = (NUMER# / DENOM#) ^ .5 ' 1/meter
' K = K / (100!) ' 1E+2 Centimeters in 1 Meter

END SUB

FUNCTION PPATRACT# (D#, R1#, R2#)

'-------------------------------------------------------

'=======================================================
' Program Name: PPATRACT
'
' Description:
'
' Author: Peter T. Robinson
'
' Date:
'
' Inputs: None
'
' Outputs: None
'
' Revision History:
' None
'
'=======================================================
'

A11# = (PartInfoA.Hamaker)
A22# = (MediumInfo.Hamaker)


IF Simulation = 1 THEN
 A33# = (PartInfoB.Hamaker)
ELSEIF Simulation = 2 THEN
 A33# = (PartInfoA.Hamaker)
END IF

 A# = ((A11# ^ .5) - (A22# ^ .5)) * ((A33# ^ .5) - (A22# ^ .5))

' A# = (1.15E-21) * 1E+11
 A# = A# * 1E+11 ' CONVERT [J]=[Nm] to [Dyne um]

C1# = .25 * (D# ^ 2) / (R1# ^ 2) + .5 * (D# * R2#) / (R1# ^ 2) + .5 * (D# / (2 * R1#)) +
R2# / R1#
C2# = .5 * D# / (R1# ^ 2) + .5 * (R2#) / (R1# ^ 2) + 1 / (2 * R1#)

CONST0# = A# / 12!

TERM1A# = -R2# * C2#
TERM1B# = R1# * (.25 * (D# ^ 2) / (R1# ^ 2) + .5 * (D# * R2#) / (R1# ^ 2) + .5 * D# /
R1#) ^ 2
TERM2# = -(R2# * C2#) / (R1# / (C1# ^ 2))
TERM3A# = 2 * C1# * (C2# / C1# - C2# * (.25 * (D# ^ 2) / (R1# ^ 2) + .5 * (D# * R2#)
/ (R1# ^ 2) + .5 * D# / R1#))
TERM3B# = .25 * (D# ^ 2) / (R1# ^ 2) + .5 * (D# * R2#) / (R1# ^ 2) + .5 * (D# / R1#)


 PPATRACT# = CONST0# * ((TERM1A# / TERM1B#) + TERM2# + (TERM3A# /
TERM3B#))


END FUNCTION

FUNCTION PPREPEL# (D#, R1#, R2#, TYP1, TYP2)
```

```
'-----------------------------------------------------

'=====================================================
' Program Name: PPREPEL
'
' Description:
'
' Author: Peter T. Robinson
'
' Date:
'
' Inputs: None
'
' Outputs: None
'
' Revision History:
' None
'
'=====================================================
'

' K# = 3.286401 * (ElectroInfo1.Concen) ^ .5
' Zeta1# = .002
' Zeta2# = .0012

IF TYP1 = 1 THEN ' A
 Zeta1# = ZetaInfoA / 1000!
ELSEIF TYP1 = 2 THEN 'B
 Zeta1# = ZetaInfoB / 1000!
END IF

IF TYP2 = 1 THEN ' A
 Zeta2# = ZetaInfoA / 1000!
ELSEIF TYP2 = 2 THEN 'B
 Zeta2# = ZetaInfoB / 1000!
END IF

H# = D# - R1# - R2#

KAP# = K / 1000000!

IF H# >= 0! AND H# THEN 'ppeter

Dielec# = (MediumInfo1.Perm) * (8.854187799999999D-12)
 COM1# = Zeta1# ^ 2 + Zeta2# ^ 2
 COM2# = 1 - EXP(-KAP# * H#)
 COM3# = 1 + EXP(-KAP# * H#)
 COM4# = KAP# * EXP(-KAP# * H#)

 CONST0# = (-Dielec# * R1# * R2# * COM1#) / (4 * R1# + 4 * R2#)

 TERM1A# = (2 * Zeta1# * Zeta2# * COM2#) / (COM1# * COM3#)

 TERM1B# = -(COM4# / COM2#) - (COM3# * COM4#) / (COM2# ^ 2)

 TERM2# = (2! * KAP# * EXP(-2! * KAP# * H#)) / (1 - EXP(-2! * KAP# * H#))

 PPREPEL# = (CONST0# * (TERM1A# * TERM1B# + TERM2#)) * 100000! ' Convert [J/m]=[N] to [Dyne]

ELSE
 PPREPEL# = 0!
END IF

END FUNCTION

SUB PRINTDATA (RECNUM, X() AS DOUBLE, Y() AS DOUBLE, Z() AS DOUBLE)

PRINT #2, USING "####"; RECNUM

FOR JJ = 1 TO NUMPART
 PRINT #2, USING "####.#####^^^^"; X(JJ);
 PRINT #2, ",";
 PRINT #2, USING "####.#####^^^^"; Y(JJ)
NEXT JJ

END SUB
```

SUB TRANSDATA

```
SUB PRINTHEADER (RECNUM, X() AS DOUBLE, Y() AS DOUBLE, Z() AS
DOUBLE, RADIUS() AS DOUBLE, COLR())

HEADER$ = "CSS DATA FILE"
PRINT #2, USING "&"; HEADER$

PRINT #2, USING "###"; NUMPART

FOR JJ = 1 TO NUMPART
 PRINT #2, USING "###.######"; RADIUS(JJ);
 PRINT #2, ",";
 PRINT #2, USING "###"; COLR(JJ)
NEXT JJ

PRINT #2, USING "####"; RECNUM

FOR JJ = 1 TO NUMPART
 PRINT #2, USING "######.######"; X(JJ);
 PRINT #2, ",";
 PRINT #2, USING "######.######"; Y(JJ)
NEXT JJ

END SUB

FUNCTION SIGN
 RANDOMIZE (TIMER)

 NUM! = RND
 IF NUM! > .5 THEN
 SIGN = 1
 ELSE
 SIGN = -1
 END IF


END FUNCTION
```

```
SUB TRANSDATA

'-----------------------------------------------------

'=====================================================
' Program Name: TRANSDATA
'
' Description:
'
' Author: Peter T. Robinson
'
' Date:
'
' Inputs: None
'
' Outputs: None
'
' Revision History:
' None
'
'=====================================================
'
'TFIL$ = "C:\QB\CSSM\CSSTEMP"
TFIL$ = "CSSTEMP"
TFILE$ = "CSSTEMP" + CHR$(0)

CALL EXIST(TFILE$, FILEXISTS%)

IF FILEXISTS% THEN

 OPEN TFIL$ FOR RANDOM ACCESS READ WRITE AS #1

 GET #1, 1, PartInfoA
 GET #1, 2, PartInfoB
 GET #1, 3, MediumInfo1
 GET #1, 4, ElectroInfo1
 GET #1, 5, FiberInfo
 GET #1, 6, MENULVL
 GET #1, 7, pHInfo1
```

ParlInfoA

```
GET #1, 8, ZetaInfoA
GET #1, 9, ZetaInfoB
GET #1, 10, ZetaInfoF
GET #1, 11, DYNMDL
GET #1, 12, Simulation
GET #1, 13, INITDAT
GET #1, 14, TimeIncr

CLOSE #1
ELSE
CLS

PRINT
PRINT
PRINT " CSS.EXE must be run first"
END IF

END SUB

SUB TRANSDATAP (FILEPAR$)

'-------------------------------------------------

'=================================================
' Program Name: TRANSDATAP
'
' Description:
'
' Author: Peter T. Robinson
'
' Date: August 1992
'
' Revision History:
' None
'
'=================================================
'

OPEN FILEPAR$ FOR RANDOM ACCESS READ WRITE AS #1

PUT #1, 1, PartInfoA
PUT #1, 2, PartInfoB
PUT #1, 3, MediumInfo1
PUT #1, 4, ElectroInfo1
PUT #1, 5, FiberInfo
PUT #1, 6, MENULVL
PUT #1, 7, pHInfo1
PUT #1, 8, ZetaInfoA
PUT #1, 9, ZetaInfoB
PUT #1, 10, ZetaInfoF
PUT #1, 11, DYNMDL
PUT #1, 12, Simulation
PUT #1, 13, INITDAT
PUT #1, 14, TimeIncr
PUT #1, 15, K#

CLOSE #1

END SUB
```

Disk # _____ _____

_____

_____

Disk # _____ _____

_____

_____

Disk # _____ _____

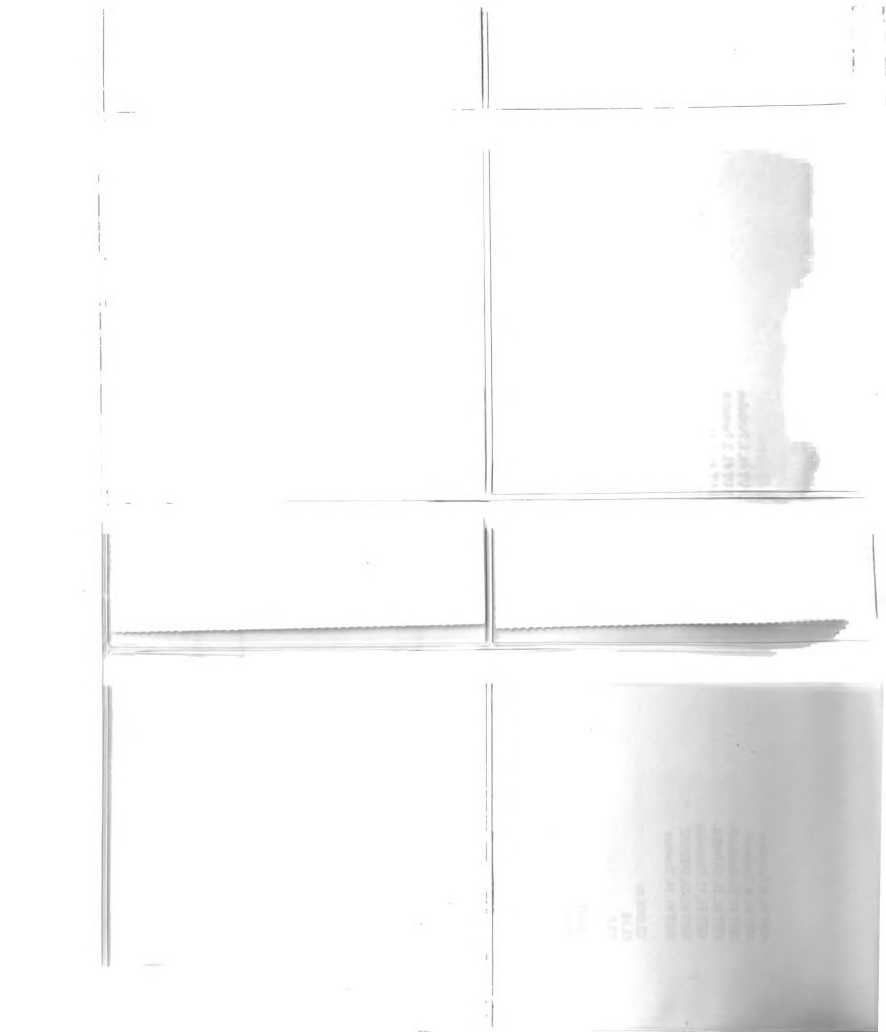
_____

_____

Disk # _____ _____

_____

_____



MICHIGAN STATE UNIV. LIBRARIES
31293008812798